



# Parallel manipulators.Part I:Theory design,kinematics,dynamics and control

Jean-Pierre Merlet

## ► To cite this version:

Jean-Pierre Merlet. Parallel manipulators.Part I:Theory design,kinematics,dynamics and control.  
[Research Report] RR-0646, INRIA. 1987, pp.68. inria-00075907

**HAL Id: inria-00075907**

**<https://inria.hal.science/inria-00075907>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# IRIA

UNITÉ DE RECHERCHE  
IRIA-SOPHIA ANTIPOLIS

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
BP 105  
78153 Le Chesnay Cedex  
France  
Tél (1) 39 63 55 11

## Rapports de Recherche

N° 646

### PARALLEL MANIPULATORS PART I: THEORY DESIGN, KINEMATICS, DYNAMICS AND CONTROL

Jean-Pierre MERLET



Mars 1987

# **PARALLEL MANIPULATORS**

**Part 1 : Theory**  
**Design, Kinematics, Dynamics and Control**

**Jean-Pierre MERLET**

ou le charme discret des univers parallèles en robotique

"la technique entre insensiblement dans sa troisième phase. La première était titanique; elle visait à édifier le monde des machines. La seconde fut rationnelle, et aboutit à l'automatisme parfait. La troisième est magique, car elle donne vie aux automates en leur donnant un sens. La technique prend un caractère d'enchantement; elle se plie aux désirs. Au rythme s'ajoute la mélodie. L'on voit s'ouvrir en même temps une nouvelle forme de l'être; nous pouvons mettre les clefs de côté."

"HELIOPOLIS", ERNST JÜNGER, 1949

Résumé

Les manipulateurs parallèles sont des robots où les différents segments ne sont pas placés successivement à partir de la base vers la "main" comme dans les manipulateurs industriels mais au contraire tous connectés directement à la fois et à la base et à la main. Ce type de robot présente une meilleure précision de positionnement ainsi qu'une grande facilité d'adaptation à la commande par retour d'efforts. Ils présentent donc un grand intérêt pour la réalisation d'opérations mécaniques (assemblage, ébavurage...) ainsi qu'une alternative aux articulations classiques utilisées dans les manipulateurs actuels. Après un bref rappel de l'évolution qui a conduit à introduire ce nouveau type d'architecture on présente les différents problèmes qui en découle. On expose ainsi les problèmes de cinématique inverse et directe, d'espace atteignable. Puis les problèmes de dynamique sont présentés ainsi que des méthodes de génération de trajectoires et de commande en force de ces manipulateurs. Des résultats originaux sont exposés en particulier pour le problème de la cinématique directe avec en particulier une formulation analytique de la matrice jacobienne. Pour conclure différents programmes sont brièvement exposés, aussi bien pour l'aide à la conception de ce type de manipulateur que pour la commande et les applications.

Summary

Parallel manipulators have a specific mechanical architecture where all the links are connected both at the basis and at the gripper of the robot. This kind of manipulator has a better positioning ability than the classical robot. We adress first the inverse and direct kinematics problems with an advance on the difficult direct problem. An analytic formulation of the jacobian matrix is proposed. We adress then the problem of the working space and we propose a new approach to establish a dynamic model of these manipulators. Motion control and Force-feedback command of these robots are studied and, as a matter of conclusion, we present some programmes which dealt with the design, the control and applications of parallel manipulators.

**Avertissement :** Cette étude des robots parallèles sera effectuée en trois parties. La première présentée ici concerne les problèmes théoriques posés par cette catégorie de manipulateurs, à l'exception de l'étude des configurations singulières, sans s'attacher à une structure particulière ou du moins en tentant de garder le maximum de généralité aux cas traités. La deuxième partie sera uniquement consacrée à l'étude des configurations singulières à l'aide de la géométrie de Grassman. La troisième fera le point sur les applications et les problèmes pratiques tels qu'ils découleront de la réalisation du prototype développé à l'INRIA.

## **1ere partie: un aperçu historique**

### **Des robots semblables aux hommes**

Le mythe fascinant de la machine crée par l'homme à son image est à la base d'un grand nombre de légendes. Sans vouloir en faire une énumération chronologique ce type de légende est courant dans l'antiquité égyptienne, grecque ou hébraïque. On peut ainsi citer le modèle d'homme dont on attribue la paternité à Ptolémée II Philadelphe, roi d'Egypte ou les jeunes filles mécaniques au service du forgeron Héphaïstos décrit dans L'Illiade d'Homère.

Ainsi l'architecte Dédale aurait construit un gigantesque robot de bronze, Talos, pour garder l'île de Crète des étrangers. S'il en capturait un, son corps se portait au rouge et il l'étreignait. Tout aussi légendaire est Galatée, statue créée par Pygmalion, dont il tomba amoureux et auquel Vénus donna la vie. Quatre siècle avant Jésus-Christ Archytas fabriqua, dit on, une colombe volante en bois alors qu'Héron d'Alexandrie fût le premier à faire reproduire par ses automates les fonctions motrices humaines. La tradition hébraïque a elle aussi de telles légendes, celles des "golem", des hommes incomplets comme celui du rabin Loew qui échappa au contrôle de son créateur.

Plus proche de nous la première matérialisation de ces légendes a été la construction d'automates n'ayant pas un rôle fonctionnel mais dont le but était la reproduction la plus fidèle possible du "vivant". Citons l'"homme de fer" du philosophe Albert le Grand dont la construction dura 30 ans. Il en fit son portier et lui donna un vocabulaire restreint. Rendant visite à son maître, Thomas d'Aquin fût si horrifié d'être confronté à une machine qu'il la démolit à coups de bâton. Au XV<sup>eme</sup> siècle l'astronome Regiomontanus aurait construit un aigle artificiel qui vola au devant de l'empereur Maximilien lorsqu'il approcha de Nuremberg. Le mécanicien Cremona construisit en 1540 une jeune fille automate jouant du luth pour exprimer son allégeance à l'empereur Charles V.

Mais la grande période des automates fût le XVIII<sup>eme</sup> siècle avec Descartes et sa

"fille Francine" , les Jacquet-Droz et leurs automates à figures humaines, la "joueuse de tympanon" de Roentgen et Kintzing et bien sûr les fameux automates de Vaucanson comme le "Joueur de Galoubet" et le célèbre "Canard". Cette grande époque des automates, objets ludiques, prit fin au début du XIX<sup>ème</sup> siècle.

L'évolution de la technologie et les besoins impérieux de l'appareil de production fit émerger une nouvelle génération de machines qui prolongaient dans un premier temps le concept des automates mais dont l'aspect utilitaire devint peu à peu prépondérant. Le mot même de robot est issu du mot tchèque qui signifie travail et même travail pénible. Karel Capek , l'auteur de la pièce de théâtre RUR, mis en scène ces esclaves mécaniques, les "robots". Mais les premiers robots électriques avait encore un aspect anthropomorphique: ils imitaient (mal ...) certaines fonctions humaines mais pour la première fois ils étaient munis de systèmes qui leur permettaient d'avoir une certaine perception de l'univers et de modifier légèrement leurs actions en conséquence. Cette évolution ne se fait pas sans mal ni sans farce comme cette histoire, probablement apocryphe, du premier robot de la "General Motors" qui se détraquait de manière aléatoire: les ingénieurs après de nombreux et vains efforts ne savaient plus où donner de la tête jusqu'au jour où l'un d'entre eux, plus observateur, remarqua que les comportements erratiques se produisaient au passages de jeunes et jolies secrétaires. N'ayant pas programmé un comportement amoureux les ingénieurs comprirent rapidement que les produits de beauté employés par ces secrétaires perturbaient la réponse de la cellule photo-électriques qui permettait au robot de garder ces distances!.

Mais ces robots ne possédaient pas assez de capacités pour être introduit dans les usines dans leurs versions anthropomorphes. La nécessité de mécanique plus appropriée aux tâches prévus a conduit à des structures plus simples. Ces architectures devaient répondre au grand besoin qui a conditionné toute la robotique moderne : aller à une position donnée dans l'espace (sous entendu: le plus près possible de ce point) en orientant la main de façon à saisir une pièce. En conséquence on a construit des mécanismes basées sur nos systèmes de référence en position : le sacro-saint repère cartésien principalement ( voir Figure 1). Mais ces robots cartésiens présentent de nombreux inconvénients comme la nécessité d'englober le volume où va évoluer la partie utile du robot c'est-à-dire sa "main".

Pour contrevenir à ces inconvénients on a cherché une structure où le volume de travail n'est pas pénalisé par l'architecture mécanique et offre un grand nombre de degrés de liberté, c'est-à-dire d'articulations permettant de placer et d'orienter la "main" du robot. Rappelons qu'il est nécessaire d'avoir au moins 6 degrés de liberté pour positionner la "main" en un point de l'espace , avec une orientation donnée. Et quel meilleur exemple d'un tel mécanisme que le bras humain avec ses 47 degrés de liberté? Mais la commande d'un grand nombre de degrés de liberté est complexe et les tâches dévolues au manipulateur industriel n'en nécessite pas tant : parfois 4 degré de liberté sont suffisants et rarement plus de 7. Ainsi naquit une deuxième génération de robot dont l'anthropomorphisme se limite à évoquer le bras humain : c'est en fait la génération des robots actuels. Notons d'ailleurs que cette tendance ne se limite pas à la seule architecture mécanique mais qu'elle intervient aussi au niveau de l'organe chargé de la saisie des pièces. De nombreux laboratoires créent des mains artificielles semblables, dans leurs principes, à la main humaine, mais avec un nombre réduit de doigts.

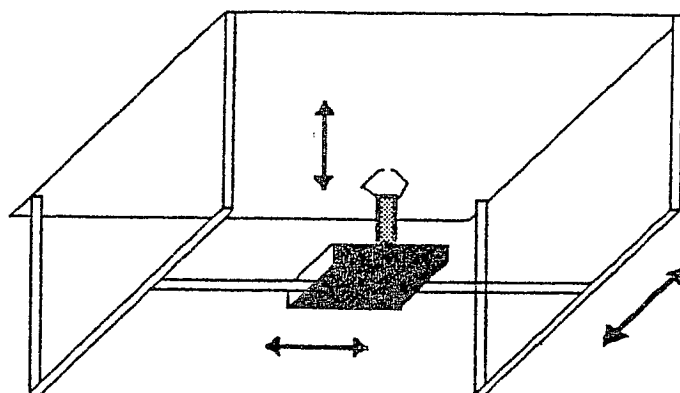


Figure 1 : robot cartésien

Et n'oublions pas les "muscles" des robots, les "actionneurs" dans la phraséologie des roboticiens, c'est-à-dire le langage des manipulateurs (noté LM par la suite), qui sont en général des moteurs électriques ou hydrauliques mais qui peuvent être aussi constitués de "muscles" pneumatiques c'est-à-dire de long fuseaux caoutchoutés creux dont la longueur varie selon leur pression interne et que l'on relie à des systèmes d'air comprimé, remplaçant ainsi la contraction de nos fibres musculaires.

Certe il y eut quelques adaptations et modifications, parfois de taille, par rapport à notre bras. La plus importante concerne la rigidité des systèmes actuels ; la tendance est en général de construire des bras les plus rigides possibles pour simplifier la tâche de l'informatique contrôlant le système, alors que notre bras possède par essence des articulations essentiellement flexibles contrôlées à l'aide de moyens sensoriels sophistiqués, et supervisées grâce à un long apprentissage. Mais dans le principe on se trouve toujours en présence du même modèle : partant d'une "épaule" on trouve une succession de "segments" reliés les uns aux autres par des articulations ne permettant le plus souvent qu'une simple rotation autour d'un axe fixe. Et c'est cette structure, assez simple dans son principe, qui pose de nombreux problèmes qui font encore l'objet d'importantes études. Les raisons des difficultés rencontrés sont de deux ordres : le premier, de fond, est l'incohérence qu'il y a entre le but exprimé et les moyens utilisés pour l'atteindre : plus simplement pour atteindre une consigne exprimée en cartésien on commande uniquement des orientations. Et ceci crée un casse-tête pour la simple raison que, si l'on sait bien passer des orientations aux positions par de simples combinaisons de fonctions trigonométriques ( en LM on dit "résoudre le problème cinématique directe"), par contre passer des positions aux orientations ( " le problème cinématique inverse" ) revient à inverser les dites combinaisons et il est facile de s'apercevoir que cela peut être un problème complexe.

Le deuxième ordre de difficulté est structurel et tient à l'aspect séquentiel des segments.

Parlons d'abord précision et illustrons ce propos par l'expérience suivante : munissez vous d'une aiguille et d'un fil, plantez l'aiguille dans un support placé à environ 50 centimètres et tentez d'enfiler votre fil dans le chas de l'aiguille. Le moindre tremblement au niveau de l'épaule se traduit par un mouvement de plusieurs millimètres au bout du bras. Si vous corrigez votre erreur de l'épaule il faut ajuster toutes les orientations des autres segments. Difficile? et bien c'est ce que l'on demande à un robot industriel. La structure séquentielle est intrinséquement médiocre pour obtenir une bonne précision de positionnement dans l'espace. Reprenons notre expérience pour illustrer une autre caractéristique des robots séquentiels : au lieu de commencer votre tentative en plaçant votre main à proximité de l'aiguille partez de la position bras tendu à 50 centimètres du chas et essayez votre chance dans un essai "au vol". Vos membres sont soumis à des effets dynamiques qui compromettent vos chances; et c'est encore pire pour un robot où les segments sont massifs (la rigidité...) ainsi que les actionneurs (partant de l'épaule ils doivent assurer le mouvement mais aussi supporter le poids des segments suivants). Cette "lourdeur" des manipulateurs classiques est assez bien illustrée par le rapport masse du manipulateur sur masse transportable qui est de l'ordre de 35 pour les manipulateurs électriques. En guise de liaison avec le chapitre suivant concluons sur un autre type de manipulateur, utilisé surtout dans les robots destinés, par exemple, à peindre les carrosseries des automobiles; ceux-ci doivent atteindre les coins les plus reculés dans des postures complexes. Ils doivent donc posséder plus que les 6 degrés de liberté strictement nécessaires et une solution proposée illustre l'imitation du vivant par sa seule appellation : les robots avec une "trompe d'éléphant" (parfois nommé aussi poignet de Fourier).

Cette introduction du vivant non-humain permet de penser que la structure du bras humain n'est pas la seule à pouvoir être utilisée en robotique.

### Des autres organes articulés

Par opposition au manipulateur séquentiel on a défini les manipulateurs parallèles: nous allons considérer deux corps, l'un supposé fixe, la base, et l'autre le mobile, qui sont reliés par des segments. Chaque segment est connecté à la fois à la base et au mobile (Figure 2).

Le terme de robot parallèle s'explique tout d'abord par le "semi" parallélisme des "jambes" du manipulateur et aussi par opposition avec l'aspect "séquentiel" des manipulateurs classiques. Nous verrons dans la suite qu'il y a d'autres justifications à cette appellation.

L'intérêt de la structure parallèle apparaît intuitivement pour ce qui concerne la précision de positionnement: en effet on voit que l'imprécision des manipulateurs, dues aux erreurs de mesure des capteurs de position, n'est plus fortement additive en terme d'erreurs de position aux articulations comme dans les manipulateurs classiques. Par exemple si l'on considère des segments parallèles, l'erreur en position du mobile selon l'axe des segments va être sensiblement bornée par l'erreur maximum des capteurs de position. En contrepartie, tout ayant un prix, l'espace balayé va être réduit pour la même raison.

Une telle structure a particulièrement intéressé des mathématiciens comme Euler et Cauchy, en particulier pour ce qui concerne la rigidité d'un tel assemblage de poutres. A



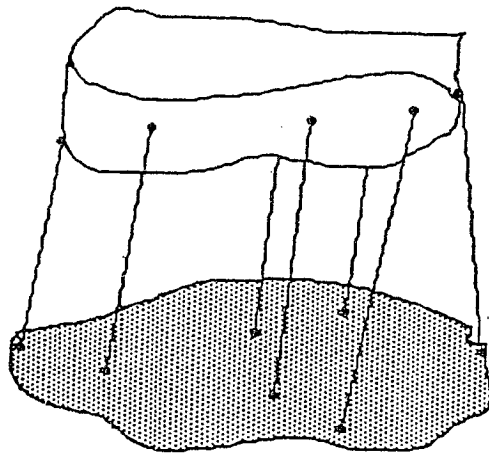


Figure 2: robot parallèle

ce titre l'étude a été intensive pour déterminer qu'elle devait être le nombre minimum de poutres pour assurer la rigidité mais aussi qu'elle en était le maximum pour que l'on puisse calculer les forces exercées sur les "jambes" en fonction des efforts exercée sur le mobile. Si on peut calculer de manière unique ces forces la structure est dite "isostatique" et si les problème compte trop d'inconnues elle est "hyperstatique". Nous reviendrons en détail sur ces notions.

L'aspect "manipulateur" est plus difficile à attribuer. En effet on voit que si l'on modifie la longueur d'une des jambes le mobile bouge (n'oublions pas que les segments sont tous articulés et qu'ils peuvent tous changer d'orientation). Si par exemple l'on place un vérin électrique dans une jambe on peut ainsi créer des mouvements. Le mouvement qui en résulte est difficilement exprimable en terme cartésien car il est composé à la fois de translations et de changements d'orientation. Les auteurs américains attribuent la première utilisation de ces robots parallèles à Stewart qui réalisa en 1965 un simulateur de vol pour les avions sur un tel principe (Figure 3). Cette paternité est très douteuse puisque dans le papier même de Stewart un commentateur (V.E. Gough) explique, photo à l'appui, qu'il avait réalisé en 1947 une machine semblable pour procéder à des essais de pneumatiques. Nous nous en tiendrons donc à notre appellation de robots parallèles.

Ces réalisations s'appuient sur les résultats des études anciennes où il est montré que pour obtenir nos 6 degrés de liberté il est nécessaire d'avoir au moins 6 segments et que pour obtenir l'isostaticité il en faut un maximum de 6. L'utilisation de ce type de robot pour les simulateurs de vol n'est pas l'effet du hasard et de telles structures sont encore couramment les seules employées dans ce domaine. Ce n'est d'ailleurs pas le seul domaine d'application puisqu'il a eu aussi un projet de simulateur pour l'apprentissage de l'équitation dont la mécanique était basée sur une telle structure. L'explication, qui sera développée ultérieurement, en est très simple: le rapport masse transportable sur masse

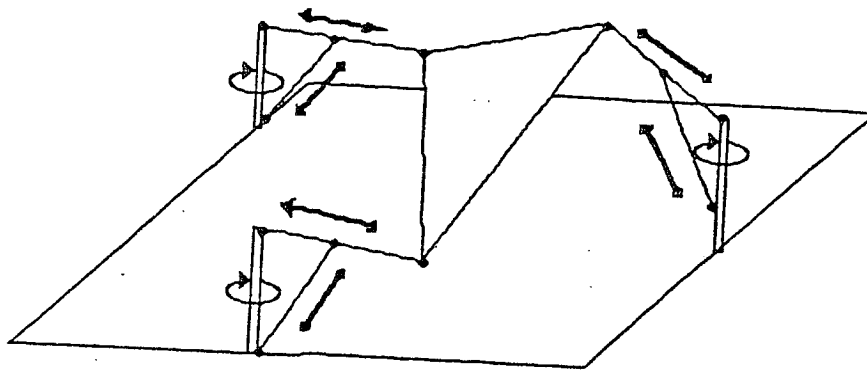


Figure 3: manipulateur de Stewart (les flèches indiquent les mouvements des segments)

du robot (qui vaut environ  $1/35$  pour les robots industriels ) est ici largement supérieur à 1 ( 90 pour le prototype de l'INRIA). La structure de Stewart n'est pas celle la plus couramment trouvée de nos jours car elle n'est que partiellement parallèle: on y trouve des actionneurs qui agissent sur la position d'autres actionneurs ( celui qui agit sur la position du vérin directement relié au mobile). Les travaux actuels s'orientent vers des manipulateurs pleinement parallèles avec une structure de base représentée sur la Figure 4.

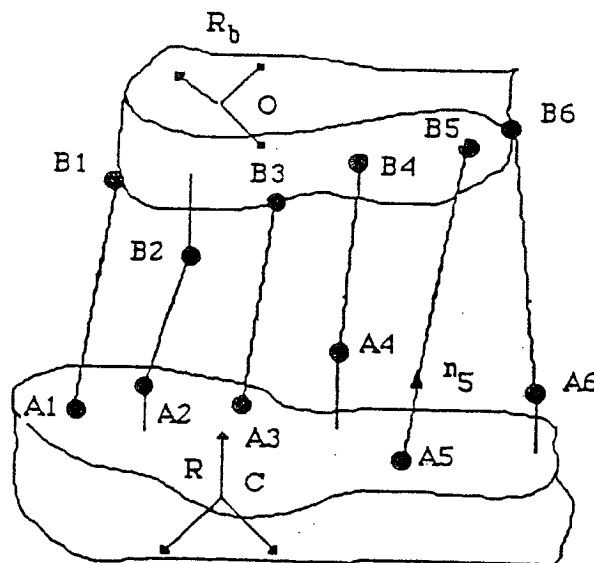


Figure 4: robot pleinement parallèle  
(un actionneur modifie l'orientation ou la longueur du segment AB)

Ce sont ces robots pleinement parallèles que nous allons maintenant étudier en détail

## Note

In order to enable a fast reading of this paper we will present in each part the problem and the results in bold characters. Original results will be denoted by a ♥

A lot of calculation (in particular the results presented in the Appendices) has been done with the aid of MACSYMA, a large symbolic manipulation program developed at MIT.

## 2nd part: Some theoritical aspects of parallel manipulators

We deal here with the study of fully parallel manipulator like the model presented in Figure 4. Basically it consists of two plates connected by 6 articulated links. In the following chapter the smaller plate will be called the mobile and the greater (which is in general fixed) will be called the base. In each link there is at least one actuator. In the case of one actuator by link we have a so-called "fully parallel" manipulator

Some manipulator of this type have been designed or studied since a long time. The first one, to the author's knowledge, was designed for testing tyres (see Mc Gough in Stewart paper). But the main use of this mechanical architecture consists in the flight simulator (see for example Stewart, Watson). The first design as a manipulator system has been done by Mac Callion in 1979 for an assembly workstation but Minski presents in the early 70's some design related to various mechanical architectures. Some other researcher have also adressed this problem: Reboulet (France), Inoue, Tanaka (as a platform for movies camera!) (Japan), Fichter, Yang, Mohamed (USA), Zamanov (Bulgaria). Even a commercial manipulator is sold by Marconi under the name of "Gadfly" for the assembly of integrated circuits. Figure 5 shows the kinematic of the link of various manipulators.

## Design considerations and parameter's choice

**Problem:** choose the number of segments, the kind of articulations for a 6 d.o.f. fully parallel manipulator

It is well known that for a mechanism whose elements present a total of  $n$  degrees of freedom, constrained by  $c$  links, which introduce  $m$  independant constraints, the number of degrees of freedom  $p$  is:

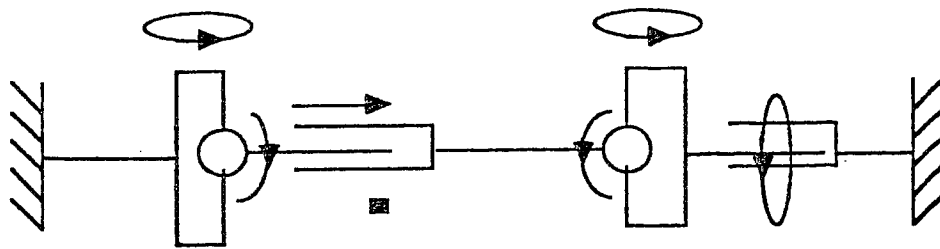
$$p = n - cm$$

Here the number of bodies is  $c+1$  ( $c$  links and one mobile) with 7 d.o.f for the links (6 d.o.f. + 1 for the actuator) and 6 d.o.f for the mobile. Thus we have:

$$p = 7c + 6 - cm$$

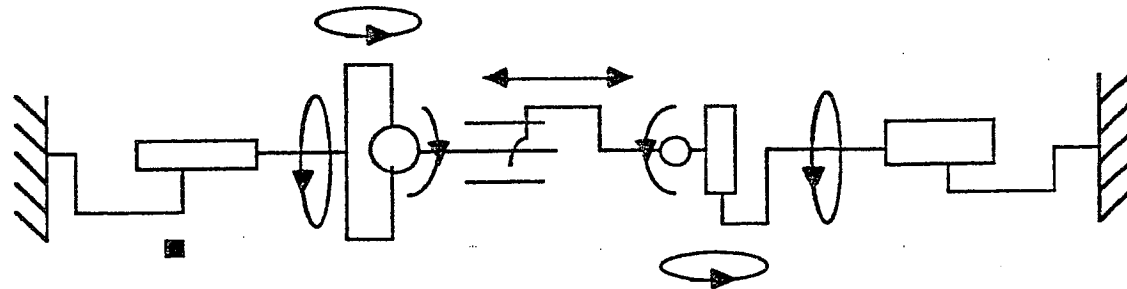
We want to have 6 d.o.f. for the mobile, thus  $p=6$  which yields to  $m=7$ . The links being articulated (and therefore constrained) on both side, the allowed couple of constraints will be (6,1), (5,2), (4,3). Technological considerations, in particular minimum clearance in the articulations whitdraw the two first couples and therefore it seems to be convenient to use a rotoïd articulation and an universal joint for the links. Thus we have determined the type of articulations.

The minimum number of links is 6 to get the 6 d.o.f. and with 6 links the system is

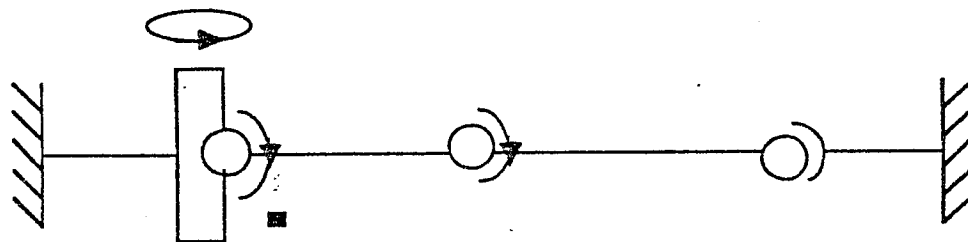


motorisation

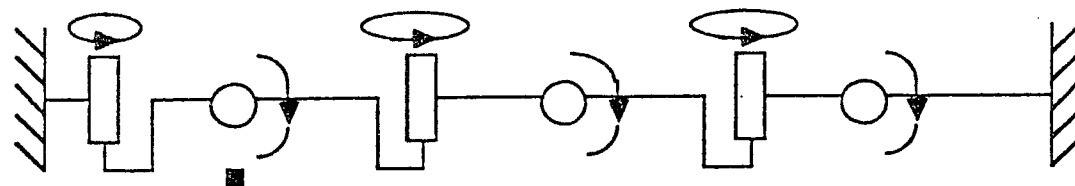
segment du RP Cert-Dera INRIA (REBOU2)



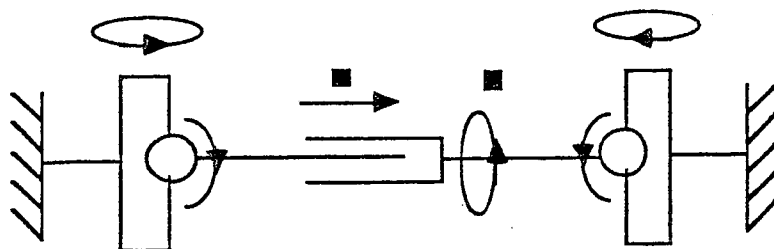
segment du RP Mac Callion (MAC 3)



Segment du RP d' INOUE (INO 2)

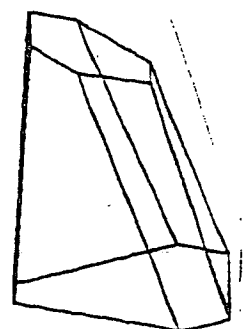


Segment du RP de MOHAMED-DUFFY (MOH 1)



segment du RP de ZAMANOV (ZAM 1)

Fig. 5 : différents segments de robots parallèles



isostatic i.e. the constraints on the link are uniquely determined if we know the forces and torques acting on the mobile. We will call the parallel robot with 6 links the minimal 6 d.o.f. fully parallel manipulator (Figure 6).

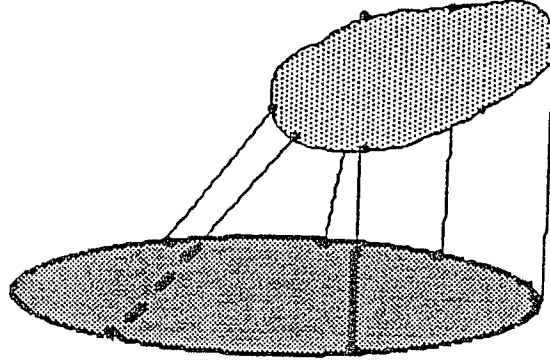


Figure 6: minimal 6 d.o.f. fully parallel robot

#### Notation

We introduce the absolute frame  $R$  with origine  $C$  and a relative frame  $R_b$  fixed to the mobile with origine  $O$  (see Fig. 6-1). The rotation matrix relating a vector in  $R_b$  to the same vector in  $R$  will be denoted by  $M$  with:

$$M = \begin{pmatrix} b1 & b2 & b3 \\ b4 & b5 & b6 \\ b7 & b8 & b9 \end{pmatrix}$$

The Euler's angles can be used with the rotation matrix  $M_e$  :

$$M_e = \begin{pmatrix} \cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\cos\theta\cos\phi & \sin\psi\sin\theta \\ \sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi & \cos\psi\cos\theta\cos\phi - \sin\psi\sin\phi & \cos\psi\sin\theta \\ \sin\theta\sin\phi & \sin\theta\cos\phi & \cos\theta \end{pmatrix}$$

with  $\psi, \theta, \phi$  the Euler's angles.

The centers of the articulations on the base for link  $i$  for the base will be denoted  $A_i$  and those on the mobile  $B_i$ . The length of link  $i$  will be noted either  $l_i$  or  $\rho_i$ , and the unit vector of this link  $n_i$ . The coordinates of  $A_i$  in frame  $R$  are  $(x_{a_i}, y_{a_i}, z_{a_i})$ , the coordinates of  $B_i$  in frame  $R_b$  are  $(x_i, y_i, z_i)$  and the coordinates of  $O$ , the origine of the relative frame,  $(x_o, y_o, z_o)$ . For the sake of simplicity the subscript  $i$  is omitted whenever

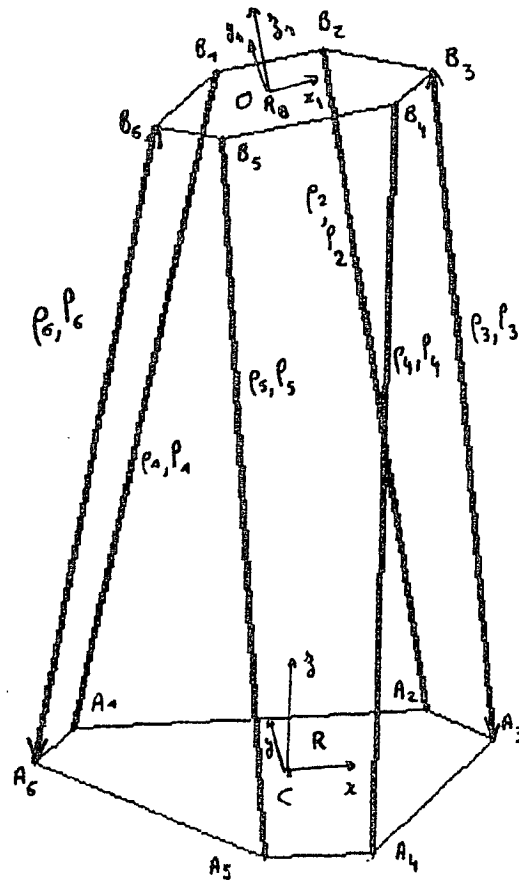


Figure 6-1: notation

it is possible and vectors will be noted in bold character. A vector whose coordinates are expressed in the relative frame will be denoted by the subscript  $r$ .

Two specific kinds of fully parallel manipulators are of particular interest. In the first one all the articulation points of both the base and the mobile lie in a plane and are symmetric along one axis ( see Figure 7). In this case, without loss of generality, we will define  $R$  such that  $z a_i = 0$  and  $R_b$  such that  $z_i = 0$ . The symmetry axis will be used as an axe of each frame  $R, R_b$ . In this case and if the articulation points lie on circles (one for the base and one for the mobile). We will call this architecture the simplified symmetric manipulator (SSM).

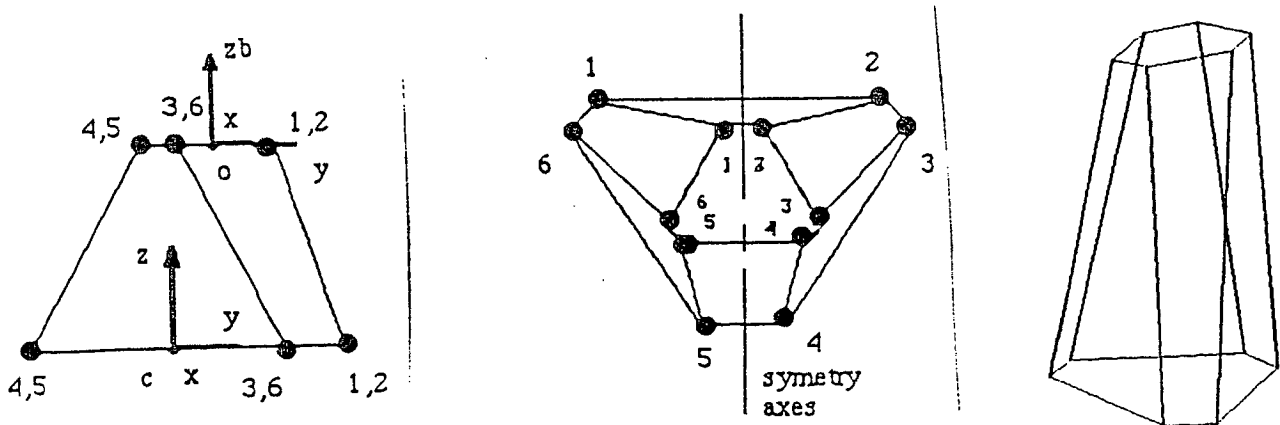


Figure 7: simplified symmetric manipulator SSM (side view, top view)

The other interesting design is presented in figure 8 : basically it is a further simplification of the simplified symmetric manipulator where some couples of articulations have

the same center of rotation: we will call this manipulator the minimal simplified symmetric manipulator (MSSM). Note that in both cases the mobile may be homothetic to the base.

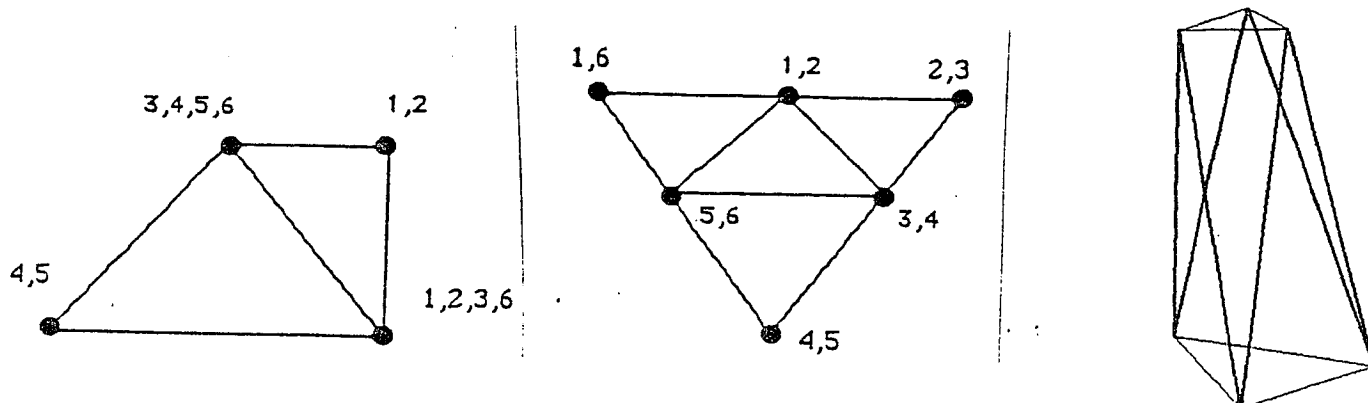


Figure 8 : the minimal simplified symmetric manipulator (MSSM)

The actuator in each link enables to control the length of each link and therefore the position and orientation of the mobile. The kinematics defines the relations between the 6 lengths and the generalized coordinates of the mobile.

An other interesting set of parameters may be choosen for the segments: the Plücker coordinates. If we define two points  $m1$  and  $m2$  belonging to a line, the Plücker coordinates of the line constitute a 6 dimensional vector  $U$ :

$$U = \left( \frac{m1m2}{||m1m2||}, \frac{Cm1 \wedge Cm2}{||m1m2||} \right)$$

where  $\wedge$  denotes the cross product and  $C$  the origine of the frame. The last part of this vector is called the moment of the line.

### Inverse kinematics problem and the inverse jacobian

**Problem :** for a known position and orientation of the mobile find the corresponding length of the links

We have:

$$AB = \rho n \quad (1)$$

and

$$AB = AC + CO + OB \quad OB = MOB_r$$

$n$  being a unit vector :

$$\rho = \|AC + CO + MOB_r\| = \|U\| \quad (2)$$

The coordinates of  $U$ ,  $(x_u, y_u, z_u)$  are given by:

$$x_u = x_o - x_a + x.b1 + y.b2 + z.b3 \quad (3)$$

$$y_u = y_o - y_a + x.b4 + y.b5 + z.b6 \quad (4)$$

$$z_u = z_o - z_a + x.b7 + y.b8 + z.b9 \quad (5)$$

Thus we have equation 6:

$$\begin{aligned} \rho = \sqrt{(x_o - x_a + x.b1 + y.b2 + z.b3)^2 + (y_o - y_a + x.b4 + y.b5 + z.b6)^2} \\ + (z_o - z_a + x.b7 + y.b8 + z.b9)^2 \end{aligned} \quad (6)$$

If the position and the orientation are given and if the positions of the articulation points are known we are able to calculate the coordinates of  $U$  and thus the length of the segment. Furthermore we notice that this calculation can be performed in a parallel way for the 6 lengths to be determined: one length is not dependant from the other five. We may also notice that the solution of the problem is unique. If we use the Euler's angles we obtain  $U$  as:

$$\begin{aligned} \rho \sin \psi_1 \sin \theta_1 &= x_o - x_a + x(\cos \psi \cos \phi - \sin \psi \cos \theta \sin \phi) - y(\cos \psi \sin \phi + \sin \psi \cos \theta \cos \phi) \\ &\quad + z \sin \psi \sin \theta \\ \rho \cos \psi_1 \sin \theta_1 &= y_o - y_a + x(\sin \psi \cos \phi + \cos \psi \cos \theta \sin \phi) + y(\cos \psi \cos \theta \cos \phi - \sin \psi \sin \phi) \\ &\quad - z \cos \psi \sin \theta \\ \rho \cos \theta_1 &= z_o - z_a + x \sin \theta \sin \phi + y \sin \theta \cos \phi + z \cos \theta \end{aligned} \quad (7)$$

with  $\psi_1, \theta_1$  the Euler's angles of the link.

We are able to calculate the unit vector  $n$  by :

$$n = \frac{U}{\rho} \quad (8)$$



If we use the matrix  $Me$  for the determination of the link's lengths the computation time on a Perkin-Elmer 3250 computer ranges between 4.6 and 5.96ms (1.73-1.77 if we tabulate the sin,cos and sqrt function) and between 1.98 and 2.33 ms on a Sun 3-75 workstation with a floating point processor.

The Plücker coordinates of the line are given by  $Up$  :

$$Up = \left( \frac{U}{\rho}, \frac{CA \wedge U}{\rho} \right)$$

### Inverse of the jacobian matrix, velocities of the segments

Let  $q$  denote the generalized coordinates of the mobile and  $J$  the jacobian matrix. In general it is convenient to use as coordinates the position of a point on the mobile and the orientation angles of the mobile (for example the Euler's angles). We have

$$\Delta q = J \Delta \rho \quad \Delta \rho = J^{-1} \Delta q \quad (9)$$

$$\rho \Delta \rho = x_u \Delta x_u + y_u \Delta y_u + z_u \Delta z_u \quad (10)$$

We are able to calculate three matrix  $K1$ ,  $K2$ ,  $K3$  such that:

$$\Delta x_u = K1 \Delta q \quad \Delta y_u = K2 \Delta q \quad \Delta z_u = K3 \Delta q$$

Thus:

$$\Delta \rho = \frac{(x_u K1 + y_u K2 + z_u K3)}{\rho} \Delta q = J^{-1} \Delta q \quad (11)$$

If we consider the derivatives of  $x_u, y_u, z_u$  a row of the matrix  $J^{-1}$  is constituted of:

$$\begin{aligned} & \frac{1}{\rho} (x_u, y_u, z_u, (x_o - x_a) \frac{dx_u}{d\psi} + (y_o - y_a) \frac{dy_u}{d\psi} \\ & (x_o - x_a) \frac{dx_u}{d\theta} + (y_o - y_a) \frac{dy_u}{d\theta} + (z_o - z_a) \frac{dz_u}{d\theta}, \\ & (x_o - x_a) \frac{dx_u}{d\phi} + (y_o - y_a) \frac{dy_u}{d\phi} + (z_o - z_a) \frac{dz_u}{d\phi} \end{aligned} \quad (12)$$

This result is quite evident for the position parameters if we notice that:

$$\frac{dx_u}{dx_o} = \frac{dy_u}{dy_o} = \frac{dz_u}{dz_o} = 1 \quad (13)$$

$$\frac{dx_u}{dy_o} = \frac{dx_u}{dz_o} = \frac{dy_u}{dx_o} = \frac{dy_u}{dz_o} = \frac{dz_u}{dx_o} = \frac{dz_u}{dy_o} = 0 \quad (14)$$

For the orientation we have to calculate:

$$\left(\left(\frac{dMe}{dq}\right)OB\right).(AC + CO + MeOB_r) \quad (15)$$

If we notice that :

$$\left(\left(\frac{dMe}{dq}\right)OB\right).MeOB_r = 0 \quad (16)$$

we get the result.

From this equation we can obtain the inverse of the jacobian matrix. Appendix 1 gives the litteral result.

Note: We have defined here the jacobian matrix  $J$  as the jacobian of the non-linear system of equations (6). Note that, in general, we choose the orientation parameters such that the jacobian matrix relates the velocities of the link to the the angular velocities of the mobile. This is not the case when we choose the Euler's angles as parameteters ( the angular velocities are not  $d\psi/dt, d\theta/dt, d\phi/dt$  ). In this case we define a new matrix  $J_v$  relating the velocities of the segments to the linear and angular velocities of the mobile. This changes only the orientation part of the former jacobian. Indeed if  $\Omega$  is the angular velocity of the mobile we have:

$$\frac{dOB_1}{dt} = \frac{dOC}{dt} + \Omega \wedge B_1C \quad (17)$$

$$\frac{d\rho_i}{dt} = \frac{dOB_1}{dt} \cdot n_1 \quad (18)$$

Therefore we have:

$$\frac{d\rho_i}{dt} = \frac{A_1B_1}{\rho_i} \cdot \left(\frac{dOC}{dt} + \Omega \wedge B_1C\right) \quad (19)$$

Therefore a row of the new jacobian matrix  $J_v$  is:

$$x_u, y_u, z_u, z.y_u - y.z_u, -z.x_u + x.z_u, y.x_u - y_u.x \quad (20)$$

The velocity of the segment i.e.  $\Delta\rho/\Delta t$  is obtained by multiplying the component of the velocity of the mobile by the inverse of the jacobian matrix. Another method is to use the Plücker coordinates and screw theory as stated by Fichter.

Every motion of a body in space can be decomposed in a rotation about one particular axis and a translation along this axis. This axis is called the instantaneous screw axis (ISA). The pitch  $h$  of the ISA is the ratio between the linear velocity  $V$  of one point belonging to the ISA and the angular velocity of the body  $\omega$ :

$$V = h\omega$$

If we know the Plücker coordinates of the ISA  $s$  and  $M$  we are able to determinate the velocity of any point on the body  $Q$  by:

$$CQ_t = V + \omega \wedge (CQ - (s \wedge M))$$

In our case we want to determine the velocity of point  $B_i$  :

$$\frac{dA_iB_i}{dt} = \mathbf{V} + \omega \wedge \left( \frac{\mathbf{U}}{\rho} - (\mathbf{s} \wedge \mathbf{M}) \right)$$

Since  $A_i$  is fixed in the base frame the above equation give the velocity of  $B_i$  in this frame. Let  $s_i$  be the first part of the Plücker coordinates of link  $i$ . The magnitude of the rate of change of length of the segment will be:

$$\frac{d\rho}{dt} = \frac{dA_iB_i}{dt} \cdot s_i$$

It is also possible to determinate the rate of change of the direction of the segment (Figure 9) by:

$$\frac{dn_i}{dt} = \frac{dA_iB_i}{dt} - \frac{d\rho}{dt} s_i$$

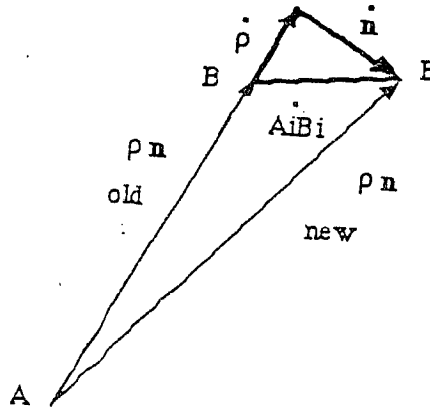


Figure 9 : change on  $A_iB_i$

The last problem is to determine the vectors  $\mathbf{s}$  and  $\mathbf{M}$  of the screw according to a given velocity  $\mathbf{V}$  of a point  $P$  of the body and  $\omega$  its angular velocity. Clearly we have:

$$\mathbf{s} = \frac{\omega}{\|\omega\|}$$

and it can be shown (see Fichter) that we have:

$$\mathbf{M} = (\mathbf{CP} - \mathbf{D}) \wedge \mathbf{s}$$

with :

$$\mathbf{D} = \frac{\mathbf{V} \wedge \omega}{(\omega \cdot \omega)}$$

**Result:** the length of each link is known analitically and is unique. Furthermore it does not depend on the length of the others links, therefore enabling a parallel computation. The computation time is less than 2.5ms. The inverse of the jacobian matrix can also be determined.

## Direct kinematics problem and the jacobian matrix

**Problem 1 :** for a set of links lengths find the corresponding position and orientation of the mobile

Note: this problem seems to be less interesting for the control than the inverse kinematics problem. However if the manipulator cannot reach a given position it is valuable to know what is its exact location.

### 1) Direct kinematic : theoretical results

Basically we have to solve the system of equations obtained in the preceding part. It can be seen that we have 6 non-linear complex equations. If we apply the rank theorem the solution will be unique except in the case where the jacobian matrix has a rank lower than 6, i.e. in the singular configuration of the manipulator. We will not deal here with singular configurations which will be treated in the second part of this paper.

The problem of direct kinematics may be illustrated by considering the following example: what is the motion of the mobile in the case where the length of only one link varies. It must be reminded that even if the length of one link does not change its orientation can change (in fact some link's orientation must change, see Cauchy's paper). The resulting motion is quite complex as it may be seen in Figure 10,11,12

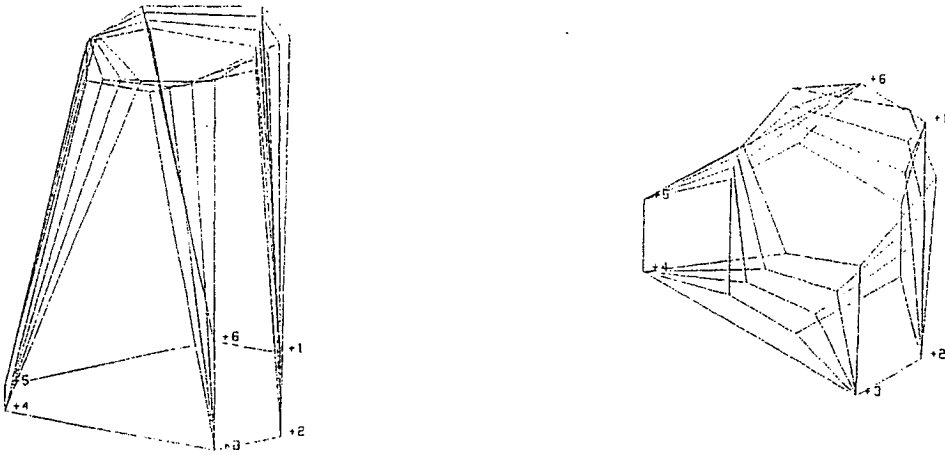


Figure 10: the motion of the mobile for variation of link1 length

Consider now the system of 6 equations in the case of the simplified symmetric manipulator (see Figure 6-1). We have:

$$\rho_i \mathbf{n}_i = \mathbf{A}_i \mathbf{C} + \mathbf{CO} + \mathbf{MOB}_{ir} \quad (1)$$

Let  $z$  and  $r$  be the radius of the circle on which lie the articulation points for the base and the mobile. Taking the norm of this equation we get

$$\rho_i^2 = z^2 + \|\mathbf{CO}\|^2 + r^2 + 2(\mathbf{A}_i \mathbf{C}^T + \mathbf{OB}_{ir}^T \mathbf{M}^T) \cdot \mathbf{CO} + 2\mathbf{A}_i \mathbf{C}^T \cdot \mathbf{MOB}_{ir} \quad (2)$$

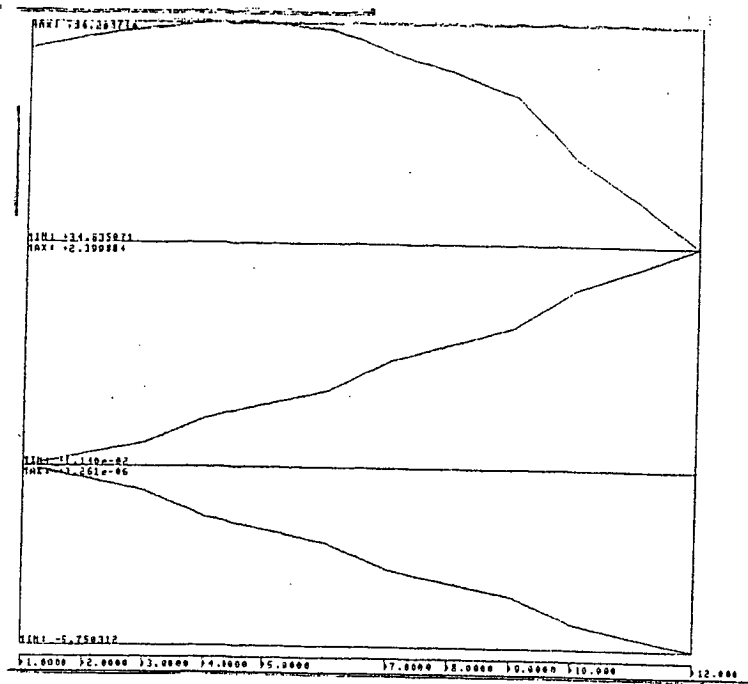


Figure 11: position of the mobile (from top z,y,x)

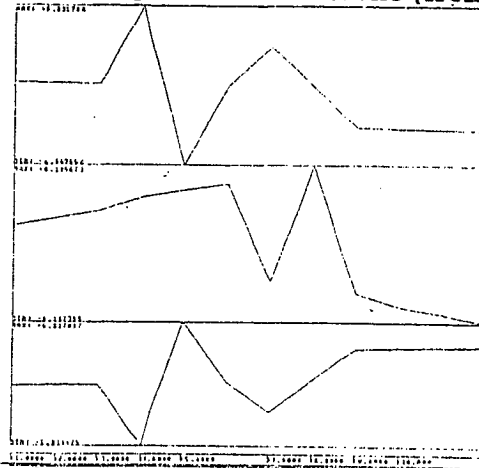


Figure 12: orientation of the mobile (from top  $\psi, \theta, \phi$ )

We make the difference between  $\rho_i$  and  $\rho_j$  for  $i \neq j$

$$\begin{aligned} \rho_i^2 - \rho_j^2 = & (2(A_i C^T + O B_{ir}^T M^T) - 2(A_j C^T + O B_{jr}^T M^T)) \cdot CO \\ & + 2A_i C^T \cdot MOB_{ir} - 2A_j C^T \cdot MOB_{jr} \end{aligned} \quad (3)$$

Thus from three equations of type (3) we can get a linear system in the three unknowns  $x_o, y_o, z_o$ .

Appendix 2 gives these three equations and  $x_o, y_o, z_o$ , expressed with the coefficients of the rotation matrix  $M$  and the difference between link 1-2,3-5,4-5.

However the remaining system is too complicated to be solved literally. We will present now a method proposed by INOUE. Consider first the following equation (4)

$$\rho_{ni} = A_i C + CO + MOB_{ir} \quad (4)$$

and assume that for symmetry reasons we have

$$\sum_{i=1}^{i=6} A_i C = \sum_{i=1}^{i=6} O B_{ir} = 0 \quad (5)$$

If we make the summation of the 6 equations (2) we get

$$\sum_{i=1}^{i=6} \rho_i \mathbf{n}_i = 6\mathbf{OC} \quad (6)$$

Thus if we know the 6 unit vectors  $\mathbf{n}$  and the links lengths we can calculate the position of the mobile. Then we can use three of equations (4) to obtain a linear system in term of  $b_7, b_8, b_9$  which enables to calculate the Euler's angle  $\phi, \theta$ . The determination of  $\psi$  from this point is rather plain.

Appendix 3 gives the details of this calculation. The main problem is to get the unit vector; INOUE use angular sensors to measure the orientations of each link and therefore calculate the vector  $\mathbf{n}$ .

## 2) Inverse kinematic : numerical method

### First order method

We use here the classical Newton method for the resolution of non-linear system of equations. Let us consider the equation:

$$\mathbf{q}_{k+1} = \mathbf{q}_k - A(I(\mathbf{q}_k) - \rho_k)$$

where  $\mathbf{q}_k$  is the generalized coordinates of the mobile at iteration  $k$ ,  $A$  a 6x6 constant matrix and  $I$  the inverse kinematics model which gives  $\rho$  for a given  $\mathbf{q}$ . Under certain assumptions this equation will converge (in particular the initial estimation must be not too "far" from the solution). A "good" matrix  $A$  is the jacobian matrix but at this time this matrix is unknown.

REBOULET computes the inverse of the jacobian matrix in a "middle" position and then perform a numerical inversion to get the jacobian in this position. The resulting matrix is considered to be a good choice for  $A$ . In fact due to the small working area of parallel manipulator it may be assumed that the jacobian matrix is constant (at least in the part where this manipulator will be used). In general the convergence will need between 2 and 5 iterations which yields a computation time range from 24 to 37 ms on a Perkin-Elmer 3250 computer and 15-23 ms on a Sun 3-75 workstation with a floating point processor.

However we have observed that the method does not converge under some bad initial estimate. This will be the case when initializing the process: no information on the actual position of the manipulator is available. In this case we use a more sophisticated algorithm combining features from Newton-Raphson, Steepest descent and Marquardt methods which try to find the parameters  $\mathbf{q}$  which minimize the following function:

$$\sum_{i=1}^{i=6} (\rho_i^2 - I(\mathbf{q})^2)$$

If we try to find the 6 parameters the computation time varies in a range of 285-1450 ms on a Perkin-Elmer 3250 computer and 168-1490 ms on a Sun 3-75 workstation with a floating point processor. At this time convergence has been observed in all cases.

It is possible to speed up the convergence of this method by using the knowledge of the position of the mobile according to the orientation and the link's length. As it may be noticed the expression of  $y_o, z_o$  are quite complicated and their numerical computation have shown that round-off error yields bad estimation of these parameters inducing a non convergence of the minimization method. However if the method converges the convergence is speed up (convergence in less than 80 ms): this is why we use the estimation of  $x_o$  which is simpler. The computation time has now a range of 176-381 ms on a Sun 3-75 workstation.

♥ The initial problem needs to solve 6 non-linear equations but can be reduced to three because the position of the mobile plate is solution of a linear system in terms of the orientation.

Numerical first order method enables to find a solution in less than 14 ms but may not converge if the initial estimation of the solution is bad. Another method is proposed which seems to find the solution in all cases but is slower.

**Problem 2 :** find the jacobian of the simplified symmetric manipulator

The jacobian matrix is defined by the relation:

$$\Delta \mathbf{q} = J \Delta \rho \quad (0)$$

Although we have a symbolic formulation for the inverse of the jacobian matrix it may be seen that a symbolic inversion is quite difficult to perform. We propose a method to solve this problem. The parameters vector  $\mathbf{q}$  may be split to two parts, one for the position  $\mathbf{X}$  and one for the orientation  $\Omega$  and we split also the jacobian matrix to two parts  $J1$  and  $J2$ , 3x6 dimensionnal matrices, defined by:

$$\Delta \mathbf{X} = J1 \Delta \rho \quad \Delta \Omega = J2 \Delta \rho \quad (1)$$

We suppose we know a function  $F$  such that:

$$\mathbf{X} = F(\Omega, \rho) \quad (2)$$

By differentiation we get :

$$\Delta \mathbf{X} = \left( \left( \frac{\partial F}{\partial \rho} \right) \right) \Delta \rho + \left( \left( \frac{\partial F}{\partial \Omega} \right) \right) \left( \left( \frac{\partial \Omega}{\partial \rho} \right) \right) \Delta \rho \quad (3)$$

We define a 3x6 matrix  $J11$  and a 3x3 matrix  $J12$  which are deduced from  $F$  by:

$$J11 = \left( \left( \frac{\partial F}{\partial \rho} \right) \right) \quad J12 = \left( \left( \frac{\partial F}{\partial \Omega} \right) \right) \left( \left( \frac{\partial \Omega}{\partial \rho} \right) \right) \quad (4)$$

From (3) we deduce:

$$J1 = J11 + J12J2 \quad (5)$$

Thus we can calculate the upper part of the jacobian matrix if we know its lower part .  
Let us consider now this lower part. We have the relation:

$$\rho_i^2 = xu_i^2 + yu_i^2 + zu_i^2 \quad 1 \leq i \leq 6 \quad (6)$$

where  $x_u, y_u, z_u$  are expressed by:

$$xu_i = x_o - xa_i + ai(\psi, \theta, \phi) \quad yu_i = y_o - ya_i + bi(\psi, \theta, \phi) \quad zu_i = z_o - za_i + ci(\psi, \theta, \phi)$$

We have seen that we use three equations of type (6) for the determination of the function F . We differentiate the remaining three equations. For the sake of simplicity we will suppose we use equations 1, 2, 3. We have:

$$E \Delta \rho = ((x_u \Delta x_u + y_u \Delta y_u + z_u \Delta z_u))_{\Omega} \Delta \Omega + ((x_u \Delta x_u + y_u \Delta y_u + z_u \Delta z_u))_{\rho} \Delta \rho \quad (7)$$

where E is the 3x6 matrix:

$$E = \begin{pmatrix} \rho_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho_3 & 0 & 0 & 0 \end{pmatrix}$$

and the subscript  $_{\Omega}(\rho)$  denotes the derivative according to  $\Omega(\rho)$ . Let

$$U = ((x_u \Delta x_u + y_u \Delta y_u + z_u \Delta z_u))_{\Omega} \quad V = ((x_u \Delta x_u + y_u \Delta y_u + z_u \Delta z_u))_{\rho} \quad (8)$$

We will first calculate U. We have

$$U = ((x_u \Delta(x_o + a) + y_u \Delta(y_o + b) + z_u \Delta(z_o + c)))_{\Omega} \quad (9)$$

Let

$$M = ((x_u \Delta a + y_u \Delta b + z_u \Delta c))_{\Omega}$$

which is equivalent to:

$$M = (((x_o - xa + a) \Delta a + (y_o - ya + b) \Delta b + (z_o - za + c) \Delta c))_{\Omega}$$

but:

$$((a \Delta a + b \Delta b + c \Delta c))_{\Omega} = 0$$

and then:

$$M = (((x_o - xa) \Delta a + (y_o - ya) \Delta b + (z_o - za) \Delta c))_{\Omega} \quad (10)$$

Thus:

$$U = M + ((x_u \Delta x_o + y_u \Delta y_o + z_u \Delta z_o))_{\Omega} \quad (11)$$



If we define  $N1 = ((x_u, y_u, z_u))$  and use (4) we obtain

$$U = M + N1J12^T \quad (12)$$

We have:

$$V = ((x_u \Delta x_o + y_u \Delta y_o + z_u \Delta z_o))_\rho$$

which yield to:

$$V = N1J11^T \quad (13)$$

Therefore

$$E\Delta\rho = (M + N1J12^T)\Delta\Omega + (N1J11^T)\Delta\rho \quad (14)$$

and

$$\Delta\Omega = (M + N1J12^T)^{-1}(E - N1J11^T)\Delta\rho = J2\Delta\rho \quad (15)$$

Thus we get

$$J2 = (M + N1J12^T)^{-1}(E - N1J11^T) \quad (16)$$

For the determination of  $J2$  we have only to invert the 3x3 matrix  $(M + N1J12^T)$ . Appendix 4 gives the matrices  $J11$  and  $(M + N1J12^T)^{-1}$ . As for the jacobian matrix  $J_v$  relating the linear and angular velocities to the velocities of the link there is only a change for the 3x3 lower part  $J2$  with

$$\Delta\omega = J2\Delta\rho \quad (17)$$

where  $\omega$  is the angular velocities vector of the mobile. We may write the inverse of this jacobian in a block form:

$$J_v^{-1} = \begin{pmatrix} U_1 & B_1 \\ U_2 & B_2 \end{pmatrix} \quad (18)$$

where  $U_1, U_2, B_1, B_2$  are 3x3 matrix. An elementary rule of block matrices algebra yield to :

$$J2 = (-(B_2 - U_2N_1^{-1}B_1)^{-1}U_2N_1^{-1} \quad (B_2 - U_2N_1^{-1}B_1)^{-1}) \quad (19)$$

We are therefore able to compute the jacobian matrix by inverting only 3x3 matrices.

♥ The jacobian can also be expressed but its formulation is quite complicated.

A problem related to the jacobian is the innacuracy of the manipulator. Indeed for a given measurement error  $\Delta\rho$  on the link's length the corresponding error on the cartesian position of the center of the mobile is calculated through the jacobian matrix and equation (0). In the general case the calculation of the jacobian is too costly to find a general formulation. However an interesting particular case can be considered : the case where the orientations are equal to zero (this is the configuration where the manipulator will be mostly used). The problem may be stated as below.

**Problem :** find the maximum innacurracy on  $x_o, y_o, z_o$  for a given innacuracy on the links lengths measurements if the base and the mobile are perfectly parallel.

In this case three equations of type 1 are used to find the relations between the variation of  $x_o, y_o, z_o$  and the variation of  $\rho$ . By differentiating we get a system of three linear equations which can be solved yielding the values of  $\Delta x_o, \Delta y_o, \Delta z_o$  with respect to the value of  $\Delta \rho_i, \Delta \rho_j, \Delta \rho_k$ . Appendix 5 gives the details of the calculation for the simplified symmetric manipulator. Interesting results can be noticed. The maximum error value on  $x_o$  is limited by:

$$\frac{\rho_m \max(|-y_{a_k} + y_{a_j} + y_k - y_j| + |-y_{a_k} + y_{a_l} + y_k - y_l| + |-y_{a_l} + y_{a_j} + y_l - y_j|) \Delta \rho}{(denom)} \quad (1)$$

$$\begin{aligned} denom = & |(x_{a_l} - x_{a_j} - x_l + x_j)y_{a_k} + (x_{a_k} - x_{a_l} - x_k + x_l)y_{a_j} \\ & + (x_{a_j} - x_{a_k} - x_j + x_k)y_{a_l} - (x_{a_l} - x_{a_j} - x_l + x_j)y_k + (x_{a_k} - x_{a_j} - x_k + x_j)y_l \\ & - (x_{a_k} - x_{a_l} - x_k + x_l)y_j| \end{aligned} \quad (2)$$

where  $\rho_m$  is the maximum value of the link's length,  $\Delta \rho$  the error measurement on the link's length, supposed to be the same for all the links and  $j \neq k \neq l$ . In the same manner the maximum error on  $y_o$  is limited by:

$$\frac{\rho_m \max(|-x_{a_k} + x_{a_j} + x_k - x_j| + |-x_{a_k} + x_{a_l} + x_k - x_l| + |-x_{a_l} + x_{a_j} + x_l - x_j|) \Delta \rho}{(denom)} \quad (3)$$

It is therefore necessary to test all the combinations of 3 links among 6 i.e. 20 combinations, to find the upper bound of the error for  $x_o$  and  $y_o$ . As a matter of example for the prototype of INRIA, where  $\Delta \rho = 0.01$  mm we found the worst case to be  $\Delta x_o = 0.33$  mm,  $\Delta y_o = 0.32$  mm. The bounds are obtained for the maximum extension of all the segments. However if the error on each link's length has the same sign, the error on  $x_o$  and  $y_o$  will be equal to zero. The maximum error on  $z_o$  is more complicated but is upperbounded by:

$$\Delta z_{omax} = \frac{\rho_m (a_{ijk}x_o + b_{ijk}y_o + c_{ijk}) \Delta \rho}{(denom.z_{omin})} \quad (4)$$

where  $a_{ijk}, b_{ijk}, c_{ijk}$  are function of the coordinates of the articulation points. For the INRIA prototype this equation yields  $\Delta z_{omax} = 0.35$  mm. If we take  $\Delta \rho = +0.01$  mm for all links we get  $\Delta z_o = 0.0094$  mm.

### Working area and design consideration

**Problem :** Find a description of the working area of the parallel manipulator

For a classical manipulator the working area may be described due to the fact that the rotations of the wrist can be neglected. For the parallel robots this is not so easy : the maximum values of the translations are strongly dependant on the orientations. However it can be noticed that in the case of fixed orientations the boundaries of the projections of

the working area on the planes xy,xz,yz are made of circular arcs. Indeed let us consider the equations of the inverse kinematic:

$$\begin{aligned}\rho_i^2 = & (x_o - xa_i + b1x_i + b2y_i + b3z_i)^2 + (y_o - ya_i + b4x_i + b5y_i + b6z_i)^2 \\ & + (z_o - za_i + b7x_i + b8y_i + b9z_i)^2\end{aligned}\quad (1)$$

which can be written as:

$$\rho_i^2 = x_o^2 + y_o^2 + z_o^2 + e_i x_o + f_i y_o + g_i z_o + h_i \quad (2)$$

The equation of the external border of the working area is obtained when one of the  $\rho_i$  reaches the maximum or minimum value of the link's length. Thus the projection on the plane xy, xz or yz of the edge of the boundary is given by the equation:

$$x_o^2 + y_o^2 + ax_o + by_o + c = 0 \quad (3)$$

which is the equation of a circle. The boundary is thus made of circular arcs. In space this gives a rather strange working area, a kind of teapot with six feet. Figure 13 shows some of these projections and Figure 14 shows a 3D reconstruction of the working area for  $\psi = \theta = \phi = 0$ . The dimensions of the teapot are about 15 cm in y,x and between 44 and 52 cm in z. Figure 15 shows some top views of the section of the working area on the x-y plane. To obtain these figures we have built sections where  $z_o$  is constant. A first trial was made by discretizing the value of  $y_o$  and looking for the maxima and minima for  $x_o$ . There is two maxima of  $x_o$  : according to equation (2) it may be seen that the maximum value of  $x_o$  is obtained when the length of at least one of the link is maximum. According to the second order of equation (2)  $x_o$  has two solutions which are symetric in case of the simplified symetric manipulator.

With the usefull help of B. Faverjon we have used polar coordinates to find the working area. Let us consider again equation 3 and introduce the polar coordinates  $r, t$  such that:

$$x_o = r \cos(t) \quad y_o = r \sin(t) \quad (4)$$

Equation (2) becomes:

$$\rho_i^2 = r^2 + r(e_i \cos(t) + f_i \sin(t)) + h_i + g_i z_o \quad (5)$$

If we define  $t1$  such that

$$\sin(t1) = \frac{e_i}{(e_i^2 + f_i^2)} \quad (6)$$

we can write:

$$\rho_i^2 = r^2 + r \sin(t + t1) + h_i + g_i z_o \quad (7)$$

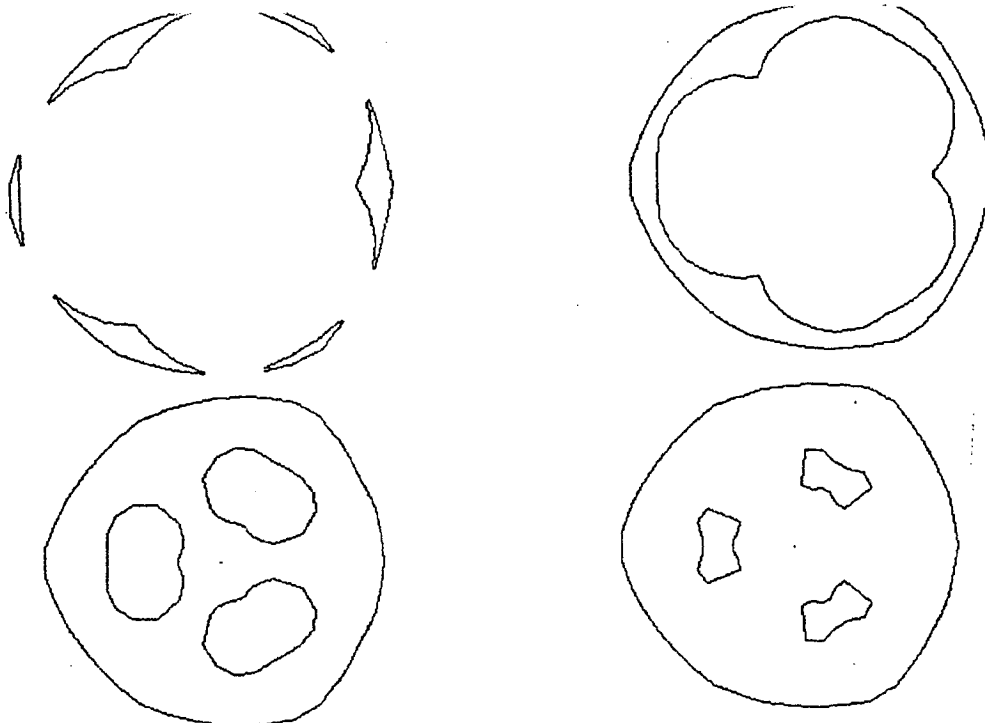


Figure 13: cross sections of the working area

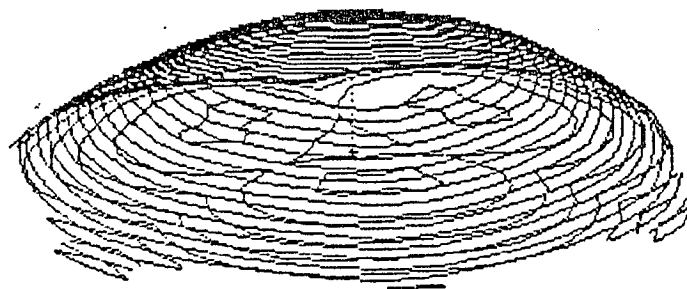


Figure 14 : 3D representation of the working area for  $\psi = \theta = \phi = 0$

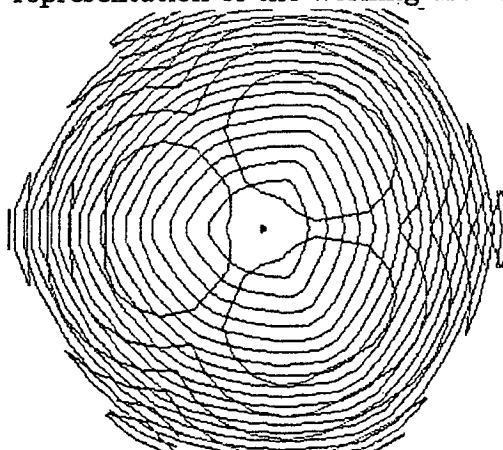


Figure 15: top views of sections of the working area // to the xy plane

For each  $t$  we are thus able to find the maxima and minima of  $r$ .

Figure 16-17 shows the working area of the same manipulator but with  $\psi = \theta = \phi = 20$  degrees.

An interesting result may be seen on these figures : the section of the working area by

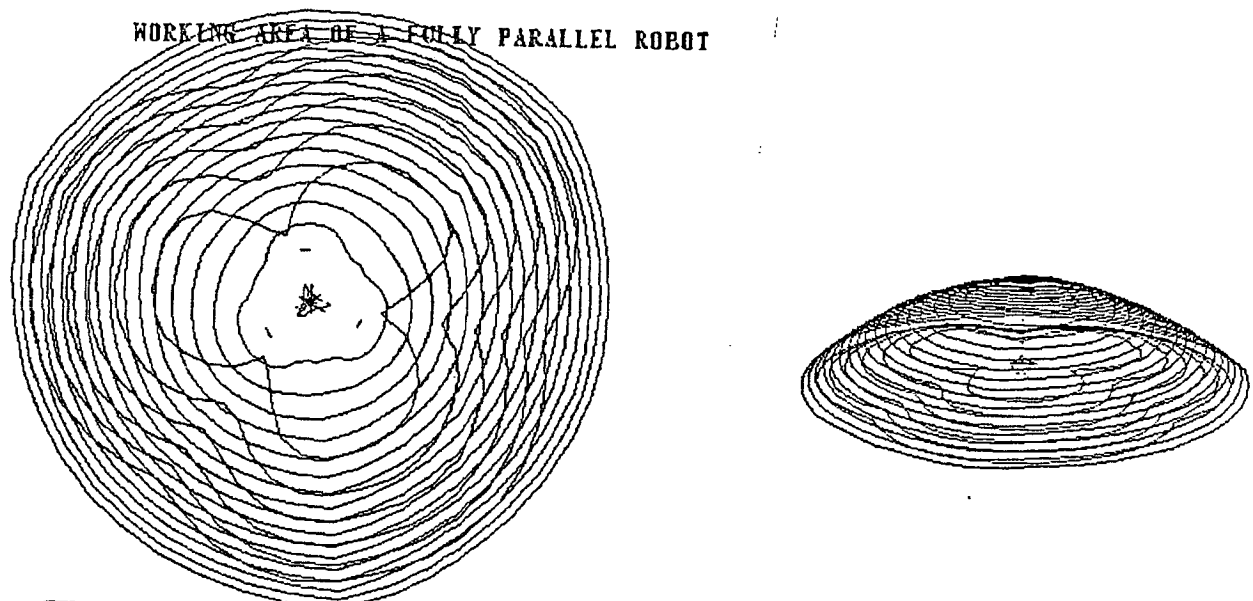


Figure 16-17 : working area for  $\psi = \theta = \phi = 20\text{degrees}$ .

a plane  $z = c^t e$  may be not connected. We may have separated "islands" where the center of the mobile can be located. This will have consequences when planning the path for the manipulator : in this case a linear motion in a  $z$  plane from one "island" to another is not possible.

♥ For a given orientation the boundaries of the projections of the working area on the plane  $xy, xz, yz$  are made of circular arcs.

## Design considerations

Let us consider now the simplified symetric manipulator in the case where  $\psi = \theta = \phi = 0$ . We may write :

$$\rho_i^2 = (x_o - xa_i + x_i)^2 + (y_o - ya_i + y_i)^2 + z_o^2 \quad (1)$$

Let us suppose that the maximum of  $\rho$  is  $lmax$  and its minimum  $lmin$ . The extrema of  $z_o$  are obtained when:

$$x_o = xa_i - x_i \quad y_o = ya_i - y_i$$

and the maximum value is  $z_o = lmax$  and the minimum is  $z_o = lmin$ . This mean that the maximum range of  $z_o$  is obtained when the links are vertical.

The maximum of  $x_o$  is obtained for  $z_o = lmin$  and has the value:

$$(xa_i - x_i) + \sqrt{l^2max - l^2min} \quad (2)$$

Its minimum is  $(xa_i - x_i)$  and therefore the variation is  $\sqrt{l^2max - l^2min}$ . Using the inequality:

$$\sqrt{l^2max - l^2min} > lmax - lmin$$

we can say that the range of  $x_o$  is greater than the range of  $z_o$ . The same reasoning can be applied for  $y_o$ , yielding the same result.

It is interesting to compare the maximum range of  $z_o$  and the ranges of  $x_o$  and  $y_o$ . In practice the parallel manipulator will be used near its mean position i.e. where  $x_o$  and  $y_o$  are equal to zero. In this case the maximum and minimum of  $z_o$  will be:

$$z_{omax} = \sqrt{l^2max - (x_i - xa_i)^2 - (y_i - ya_i)^2} \quad (3)$$

$$z_{omin} = \sqrt{l^2min - (x_i - xa_i)^2 - (y_i - ya_i)^2} \quad (4)$$

The maximum and minimum of  $x_o$  are:

$$x_{omax} : (xa_i - x_i) + \sqrt{l^2max - (y_i - ya_i)^2} \quad (5)$$

$$x_{omin} : (xa_i - x_i) + \sqrt{l^2min - (y_i - ya_i)^2} \quad (5)$$

The maximum and minimum of  $y_o$  are obtained by permuting the  $x$  and  $y$  in the preceding equations. For the INRIA prototype we have  $54cm \leq z_o \leq 59cm$ ,  $-10cm \leq x_o \leq 10cm$ ,  $-10cm \leq y_o \leq 10cm$  for a maximum variation of the link's length of 5 cm.

To obtain the maximum displacement in the vertical direction we notice that the links must be as vertical as possible. However the links are not perfect line due to the motors, force sensors, etc.... It is thus interesting to test if, for a given configuration of the articulation points on the mobile and the base some collision between all the elements of the links may happen, whatever is the position of the manipulator. We have developped a

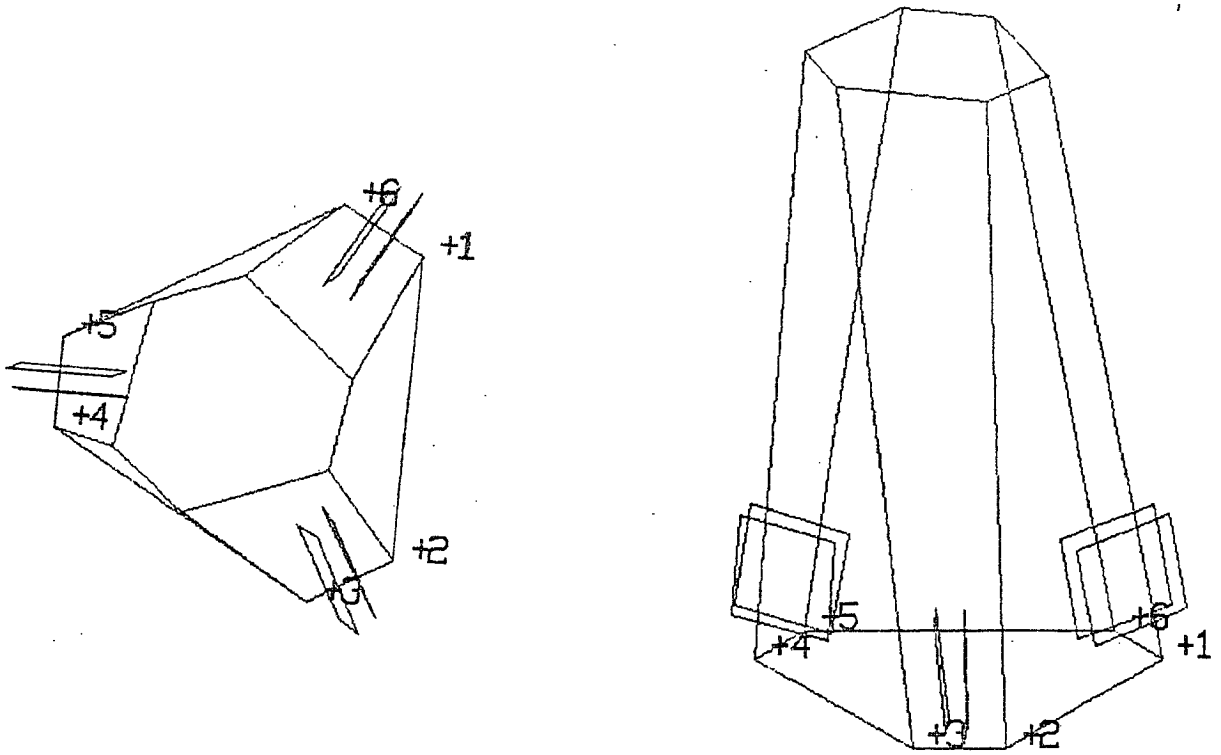


Figure 18 : intersection test (the squares must not intersect)

programm which look for an intersection of a cuboid and cylinder linked to the segment, based on hierarchical CAD program designed by B. Faverjon. Each component of the generalized coordinates of the mobile is discretized and the distance between the links is calculated at each step. The complete test for all the working area is rather long but need to be performed only one time. Figure 18 shows a graphical display of one of this test. On this figure we may see squares which are part of an electrical ram and are not allowed to intersect.

### Parallel robot as a force sensor

**Problem 1 :** For a set of forces acting on the link what is the forces and the torques acting on the mobile.

We have seen that a fully parallel manipulator with 6 links is isostatic i.e the problem as an unique solution (as its reciprocal problem). It is therefore interesting to have force sensors measuring the traction-compression forces in each link. To the author's knowledge it exists two fully parallel manipulator including force sensors in each link. The earlier was designed by C. Reboulet (CERT-DERA, Toulouse) and the second is the prototype of INRIA.

For given position and orientation of the mobile we have seen that we are able to calculate the unit vector  $n$  of each link. The mobile being in equilibrium we can write the system B:

$$F = - \sum_{i=1}^{i=6} f_i n_i \quad (1)$$

$$\mathbf{T} = - \sum_{i=1}^{i=6} \mathbf{MB}_i \wedge f_i \mathbf{n}_i \quad (2)$$

Where  $\mathbf{F}$  is the force vector acting on the mobile and  $\mathbf{T}$  the torques with respect to the point M. These equations enable us to calculate  $\mathbf{F}$  and  $\mathbf{T}$  according to the force measurements. The computation time for this procedure is about 7,2 ms on a PE 3244 computer.

An interesting application of this calculation is the determination of the position of the center of mass of an object. The preceding system gives two equations for the three unknowns and it is therefore necessary to know one of this unknown (e.g. the height of the center of mass with respect to its face lying on the mobile) or to perform the measurements in two different locations of the manipulator.

**Problem 2 :** For a generalized force vector what are the forces acting on each link?

The system B is a 6-dimensional linear system in term of the 6 unknowns  $f_i$  : therefore this problem can be solved except in the case where the manipulator is in one of its singular configuration.

However in this case the resolution of this system is difficult. It is well known that we have:

$$\mathbf{f} = J_v^T \mathbf{F}$$

where  $J_v$  is the jacobian matrix relating the generalized velocities of the mobile to the velocities of the links. Which means that we need the jacobian matrix to solve the system. We have seen that this matrix has a complicated form. It is thus more suitable to use a numerical method to obtain the solution. An iterative procedure is used by REBOULET and seems faster than numerical method solving system of linear equations. We have also implemented a method to get the jacobian matrix using block matrix algebra and it takes about 12 ms to obtain the forces acting on the link.



## Force feedback control of parallel manipulators

As we have seen before parallel manipulators are designed to be fine positioning devices with a rather small working area but also to be a force sensor. To control the parallel manipulator in tasks like assembly we need force-feedback control. An extensive study of this topic has been made and it seems that the most promising way is the so called "hybrid control" method. In a given task frame the configuration space is divided in two orthogonal varieties in which the parameters are either position or force controlled. Basically the operator indicates in the task frame which directions will be force (position) controlled. This choice is obtained through a selection matrix  $S$  such that  $S_{ii} = 1$  if the corresponding direction is force controlled (Fig 19)

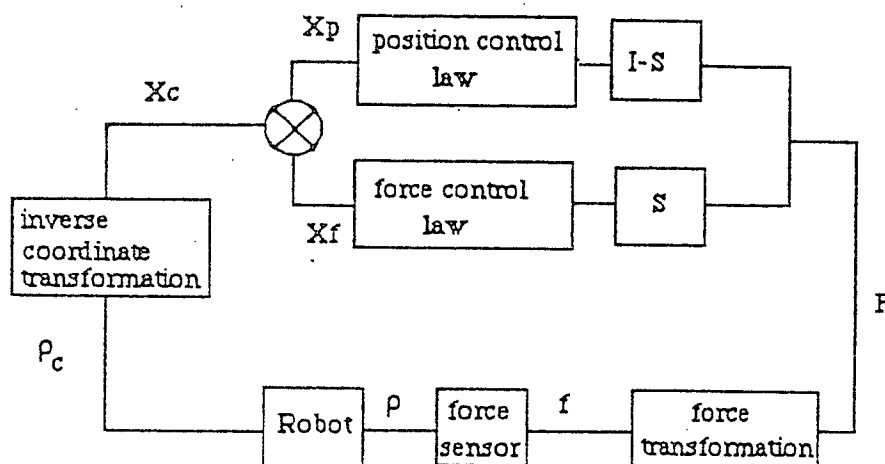


Figure 19: basic hybrid controller

This simple hybrid controller suffers from the fact that the determination of the direction to be force or position controlled is not easy. Furthermore these directions can change during a task (e.g. for the contour following in the case of a step) or may be corrupted by uncertainties (e.g. the insertion axis for an assembly task). The author has proposed an improvement of this hybrid controller based on C-surface theory. Basically in the parameters space (configuration space) the C-surface is the surface where the robot is constrained by its surrounding. The robot (a point in the configuration space) must slide on the C-surface and furthermore a motion along the C-surface normal enables to control the force magnitude. In terms of hybrid control the C-surface normal is the direction where the robot must be force controlled and the robot is position controlled in the tangent hyperplane.

It can be shown (c.f. Merlet1) that the C-surface normal can be obtained through force measurements. The formulation of the C-surface normal can be obtained rather

easily through the virtual work principle. For example if the parameters are the cartesian coordinates and the Euler's angle the 6-dimensionnal C-surface normal vector can be expressed with the measured forces and torques and two of the Euler's angles.

In the case of hybrid control, through the C-surface normal we get the actual direction to be force controlled, which will be estimated at each sampling interval of the force measurement. The motion in the tangent hyperplane (position control) is obtained through the positionnal constraints of the task. Thus we need a normal "expert" which calculates the C-surface normal according to the force measurements and generates a motion along this normal to control the magnitude of the force. Then we have a sliding "expert" which calculates the motion in the C-surface hyperplane. Figure 20 shows the resulting hybrid controller.

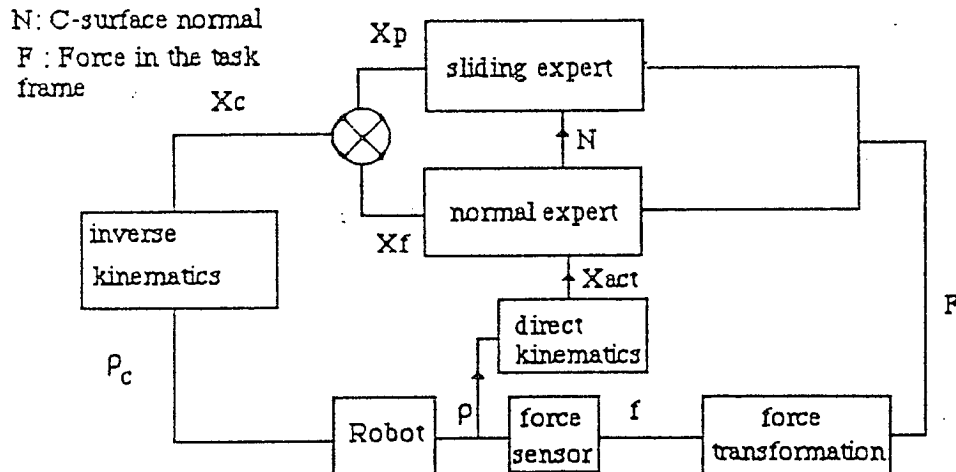


Figure 20: hybrid controller

This hybrid controller has been implemented and experiences have been conducted for the contour following problem and it gives very good results. For assembly tasks simulation has been performed and give good results. In the case of the parallel manipulator let us estimate the computation time :

	force transformation	direct kinematics	sliding
ms	7	14	1.6
	normal expert	inverse kinematics	Total
ms	1.05	1.5	25.15

This time can be improved for the parallel manipulator. In this case let us choose as parameters of the configuration space the links lengths. It can be shown that the C-surface normal is directly the measured link's forces. Indeed any allowable displacement  $\Delta p$  lie on

the C-surface tangent hyperplane but using the virtual work principle we may write:

$$\Delta \rho \cdot \mathbf{f} = 0$$

and thus the force vector  $\mathbf{f}$  is orthogonal to the tangent hyperplane. Consequently this force vector is directed along the C-surface normal.

We will have a motion  $\rho_F$  proportional to the links forces. Two cases can be considered for the generation of  $\rho_F$ : the simplest is to have a stiffness matrix  $c'_p$  expressed in the articular frame, the second is to choose a stiffness matrix  $c_p I$  in the cartesian frame such that the corresponding motion to  $\rho_F$  will be  $x_F$ . We have :

$$x_F = c_p \mathbf{F} \quad \rho_F = c'_p \mathbf{f}$$

but

$$\mathbf{F} = (J^T)^{-1} \mathbf{f}$$

and therefore

$$x_F = J \rho_F = c_p (J^T)^{-1} \mathbf{f} = J c'_p \mathbf{f}$$

which yield

$$c'_p = J^{-1} c_p (J^T)^{-1}$$

If  $c_p = c_p I$  then  $c'_p = c_p J^{-1} (J^T)^{-1}$

For the parallel manipulator it is easy to calculate  $c'_p$  because we have a symbolic formulation of  $J^{-1}$ . The sliding expert generates a motion in the tangent hyperplane according to a vector  $\mathbf{U}$  such that:

$$\rho_{act} + \rho_F + \mathbf{U} = \rho_g$$

Where  $\rho_{act}$  is the actual actuator positions and  $\rho_g$  the goal. Let  $\mathbf{v}_i$  be a basis of the tangent hyperplane. We have:

$$\mathbf{U} = \sum_{i=1}^{i=6} l_i \mathbf{v}_i$$

This coefficient  $l_i$  can be determined by:

$$l_i = (\rho_g - \rho_{act} - \rho_F) \cdot \mathbf{v}_i$$

In fact the real motion in the tangent hyperplane is  $\beta \mathbf{U}$ . The coefficient  $\beta$  can be determined by controlling either the actuator or the cartesian velocity. In the former case let  $t$  be the time between two calculations of the command and  $V_{max}$  the maximum velocity of the actuator. We get:

$$|\rho_F + \beta \mathbf{U}| = \frac{V_{Max}}{t}$$

which enables to calculate  $\beta$

In the case where the cartesian velocity is constant we will have a constant displacement between two motions with amplitude  $dep$ . This displacement is due to the articular motion  $\Delta\rho$  :

$$\Delta\rho = \rho_F + \beta U = (1 - \beta)\rho_F + \beta(\rho_g - \rho_{act})$$

The resulting displacement is  $\Delta X = J\Delta\rho$  and its amplitude is :

$$\Delta X^T \Delta X = \Delta\rho^T J^T J \Delta\rho = dep^2$$

and thus:

$$dep^2 = \beta^2((\rho_g - \rho_{act})^T J^T J (\rho_g - \rho_{act}) - \rho_F^T J^T J \rho_F - 2\rho_F^T J^T J U) + 2\beta\rho_F^T J^T J U + \rho_F^T J^T J \rho_F$$

which enables to calculate  $\beta$ . An interesting case may be considered if the orientation angles are small and  $c'_p = c_p J^{-1}(J^T)^{-1}$ . We get:

$$\beta^2 = \frac{(dep^2 - c_p^2 f^T (J^T J)^{-1} f)}{((\rho_g - \rho_{act})^T J^T J (\rho_g - \rho_{act}) - c_p^2 f^T (J^T J)^{-1} f)}$$

The main problem is that we need the product  $J^T J$ . However due to the limited working area it can be considered (as in the direct kinematic problem) that the jacobian matrix is constant. Another point is to assume small variations around a nominal position. We can then compute off line the jacobian matrix for this nominal position and consider it as a constant for the task. The resulting hybrid controller can be seen on Figure 21.

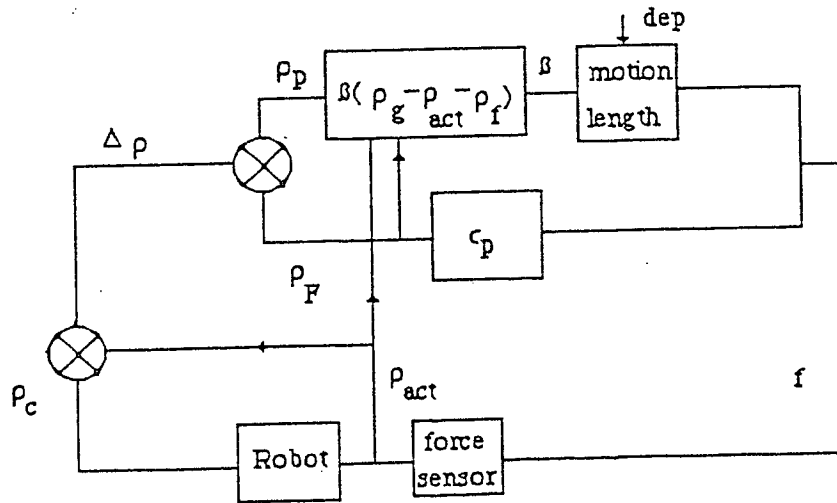


Figure 21 : hybrid controller in articular coordinates

## Parallel manipulator as a passive compliance device

In each segment of a parallel manipulator we have some elasticity due to the mechanical components. In the case of the INRIA prototype we have elements with very high rigidity (like the electrical ram used to modify the length of the links) and added with elastic dampers : these dampers are used at first to protect the force sensor, but give also a pure passive compliance to the manipulator which can be usefull in tasks like assembly for example (Merlet 86). Let us calculate the compliance matrix of this manipulator. We have:

$$\mathbf{f} = k\Delta\rho \quad \mathbf{F} = (\mathbf{J}^T)^{-1}\mathbf{f} \quad (1)$$

where  $k$  is the elasticity of the damper. But:

$$\Delta\rho = \mathbf{J}^{-1}\Delta\mathbf{X} \quad (2)$$

which yield:

$$\mathbf{F} = k(\mathbf{J}^T)^{-1}\mathbf{J}^{-1}\Delta\mathbf{X} \quad (3)$$

the compliance matrix  $\mathbf{K}$  can be therefore written as:

$$\mathbf{K} = k\mathbf{J}\mathbf{J}^T \quad (4)$$

Note that we will use here the second form of the jacobian matrix i.e.  $\mathbf{X}$  is defined as:

$$\begin{aligned} \mathbf{X} = (x_o, y_o, z_o, \frac{d\theta}{dt}\cos\psi + \frac{d\phi}{dt}\sin\theta\sin\psi, \\ \frac{d\theta}{dt}\sin\psi - \frac{d\phi}{dt}\sin\theta\cos\psi, \frac{d\psi}{dt} + \frac{d\phi}{dt}\cos\theta) \end{aligned} \quad (5)$$

The last three components of  $\mathbf{X}$  are the angular velocities of the mobile. This choice is justified by the fact that the compliance matrix relates the variation of the generalized velocities to those of the generalized forces vector.

In tasks like assembly the parallel manipulator will be used in a "central" position where  $x_o = y_o = \psi = \phi = \theta = 0$ . It is therefore interesting to calculate this matrix for this nominal position of the manipulator.

For the MSSM ROBERT has shown that it is possible to calculate this jacobian matrix at the nominal position and further calculations have shown that the compliance matrix is then diagonal. This result is very interesting for the use of parallel manipulators as passive compliance devices: it is very important that the system is decoupled so that the correction of the angular misalignments does not generate lateral errors and vice-versa. This means that a pure force generate only a translation and a pure torque only a rotation, see Merlet86. The compliance matrix can be written as :

$$\mathbf{K} = \begin{pmatrix} \frac{u}{3a^2k} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{u}{3a^2k} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{u}{24h^2k} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{u}{a^2h^2k} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{u}{a^2h^2k} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2u}{a^4k} \end{pmatrix} \quad (6)$$

where  $h$ : position on a vertical axis of the mobile,  $a$ : distance between two articulation points and  $u = 4h^2 + a^2$ . For the INRIA prototype we have  $a=22.5\text{cm}$ ,  $h = 51.4\text{cm}$  and  $k= 800 \text{ N/cm}$  which yield:

$$k_x = k_y = 0.0091\text{cm}/N \quad k_z = 0.0002184\text{cm}/N$$

We notice that the stiffness in the  $z$  direction is 41 time the stiffness in the  $x,y$  direction : such stiffnesses are very convenient for assembly task.

In the case of the SSM the compliance matrix is no longer diagonal. The form of either the compliance or stiffness matrix is:

$$K = \begin{pmatrix} u1 & 0 & 0 & 0 & u2 & u3 \\ 0 & u1 & u5 & u6 & 0 & 0 \\ 0 & u5 & u7 & u8 & 0 & 0 \\ 0 & u6 & u8 & u9 & 0 & 0 \\ u2 & 0 & 0 & 0 & u9 & u11 \\ u3 & 0 & 0 & 0 & u11 & u10 \end{pmatrix} \quad (7)$$

For the INRIA prototype we get:

$$K = \begin{pmatrix} 0.0263 & 0 & 0 & 0 & 0.000298 & 0.0000227 \\ 0 & 0.0263 & 0.0 & -0.000295 & 0 & 0 \\ 0 & 0.0 & 0.000213 & 0.0 & 0 & 0 \\ 0 & -0.000295 & 0.0 & 0.000011 & 0 & 0 \\ 0.000298 & 0 & 0 & 0 & 0.000011 & 0.0 \\ 0.0000227 & 0 & 0 & 0 & 0.0 & 0.000230 \end{pmatrix}$$

and the stiffness matrix is:

$$S = \begin{pmatrix} 55.1 & 0 & 0 & 0 & -1515.7 & -3.97 \\ 0 & 55.1 & 0.0 & 1520.1 & 0 & 0 \\ 0 & 0 & 4689.2 & 0.0 & 0 & 0 \\ 0 & 1520.1 & 0.0 & 134005.5 & 0 & 0 \\ -1515.7 & 0 & 0 & 0 & 134005.5 & 0.0 \\ -3.97 & 0 & 0 & 0 & 0.0 & 4365.5 \end{pmatrix}$$

Appendix 6 gives the symbolic formulation of the stiffness matrix for a SSM in the case where  $x_o = y_o = \psi = \theta = \phi = 0$  and if  $\psi = \theta = \phi = 0$  but  $x_o \neq 0$  and  $y_o \neq 0$ . We notice that the stiffnesses on the  $x,y$  axis are about 27 time less than the stiffness on the  $z$  axis. The matrix  $K$  can be modified by changing the vertical position of the mobile. The preceding matrix was calculated with all the links at their minimum value. If we move to their maximum extension we get the new compliance matrix:

$$K = \begin{pmatrix} 0.0326 & 0 & 0 & 0 & 0.00033 & 0.000028 \\ 0 & 0.0326 & 0.0 & -0.00033 & 0 & 0 \\ 0 & 0.0 & 0.000212 & 0.0 & 0 & 0 \\ 0 & -0.00033 & 0.0 & 0.000011 & 0 & 0 \\ 0.00033 & 0 & 0 & 0 & 0.000011 & 0.0 \\ 0.000028 & 0 & 0 & 0 & 0.0 & 0.000281 \end{pmatrix}$$

and the new stiffness matrix:

$$S = \begin{pmatrix} 45.0 & 0 & 0 & 0 & -1389.6 & -3.25 \\ 0 & 45.0 & 0.0 & 1393.8 & 0 & 0 \\ 0 & 0 & 4709.7 & 0.0 & 0 & 0 \\ 0 & 1393.8 & 0.0 & 135721.0 & 0 & 0 \\ -1389.6 & 0 & 0 & 0 & 135721.0 & 0.0 \\ -3.25 & 0 & 0 & 0 & 0.0 & 3557.5 \end{pmatrix}$$

## Dynamic model of parallel manipulators

**Problem:** find the equation of motion of parallel manipulators

### 1) Simplified approach

We will begin this part with a simplified model where we suppose that the links have no weight. In this case let us define the inertial matrix  $M$  by:

$$M = \begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & J_x & 0 & 0 \\ 0 & 0 & 0 & 0 & J_y & 0 \\ 0 & 0 & 0 & 0 & 0 & J_z \end{pmatrix}$$

We look for the free motion around a nominal position  $X_0$ . Let  $X = X_0 + \Delta X$  and  $\tau$  be the vector of link's forces. We have:

$$\tau = J^T F \quad (1)$$

where  $J$  is the jacobian matrix at  $X_0$ . For a motion  $\Delta X$  we have a corresponding motion of the links  $\Delta \rho$  generating forces in the link such that:

$$\tau = -k \Delta \rho = -k J^{-1} \Delta X \quad (2)$$

Thus the forces acting on the mobile are:

$$F = -k (J^T)^{-1} J^{-1} \Delta X \quad (3)$$

The equation of motion may be written as:

$$M \frac{d^2 \Delta X}{dt^2} = -k (J^T)^{-1} J^{-1} \Delta X = -S \Delta X \quad (4)$$

$S$  being a diagonal matrix for the MSSM we get the natural frequencies of the parallel manipulator by:

$$\begin{aligned} f_x = f_y &= \frac{\sqrt{k_x/m}}{2\pi} & f_z &= \frac{\sqrt{k_z/m}}{2\pi} \\ f_{\omega x} = f_{\omega y} &= \frac{\sqrt{k_{\omega x}/m}}{2\pi} & f_{\omega z} &= \frac{\sqrt{k_{\omega z}/m}}{2\pi} \end{aligned}$$

For the INRIA prototype where  $m = 0.25$  kg we get:

$$f_x = f_y = 3.33 Hz \quad f_z = 21.53 Hz$$



If we use the jacobian matrix of a SSM we get for the INRIA prototype:

$$f_x = f_y = 1.96Hz \quad f_z = 21.53Hz$$

## 2) Through a complete dynamic model

The preceding study was a first approach to find a dynamic model. For a complete model the mass of the links cannot be neglected: in fact for the INRIA prototype the links far outweigh the mass of the mobile.

This is why we propose a new approach based on Lagrange's formulation. The main difficulty in this approach is to find the kinetic energy of the system. We will make here two reasonable assumptions: we will suppose first that the moment of inertia of the link around axis perpendicular to the main axis are identical and, secondly, that the moment of inertia around the main axis is equal to zero.

Let  $E_m$  be the kinetic energy of the mobile. We have:

$$\begin{aligned} 2E_m = m \left( \frac{dx_o}{dt}^2 + \frac{dy_o}{dt}^2 + \frac{dz_o}{dt}^2 \right) & J_x \left( \frac{d\theta}{dt} \cos\psi + \frac{d\phi}{dt} \sin\theta \sin\psi \right)^2 \\ & + J_y \left( \frac{d\theta}{dt} \sin\psi - \frac{d\phi}{dt} \sin\theta \cos\psi \right)^2 + J_z \left( \frac{d\psi}{dt} + \frac{d\phi}{dt} \cos\theta \right)^2 \end{aligned}$$

Thus the kinetic energy of the mobile can be easily expressed in term of the position and orientation parameters. Let us consider now the kinetic energy of the links  $E_s$ . We have:

$$2E_s = J_s \left( \frac{d\theta_1}{dt}^2 + \frac{d\psi_1}{dt}^2 \sin^2\theta_1 \right)$$

where  $J_s$  is the moment of inertia of the segment around an axe perpendicular to its main axis,  $\theta_1, \psi_1$  the Euler's angle of the segment. The main problem is to expressed  $E_s$  according to the parameters. Let us remember that we have the unit vector of the link n:

$$\mathbf{n} = \begin{pmatrix} \sin\psi_1 \sin\theta_1 \\ -\cos\psi_1 \sin\theta_1 \\ \cos\theta_1 \end{pmatrix}$$

and that we can express this vector according to the parameters (see the chapter on inverse kinematics). If we calculate the derivative of  $\mathbf{n}$  and consider its norm it is easy to see that we have:

$$\left\| \frac{d\mathbf{n}}{dt} \right\|^2 = \frac{d\theta_1}{dt}^2 + \frac{d\psi_1}{dt}^2 \sin^2\theta_1$$

Thus we can calculate the kinetic energy of the system according to the parameters. However a complete expression of the dynamic model is rather complicated. We are currently investigating further developments.

## Conclusion

We have presented some considerations on the features of parallel manipulator. This kind of architecture seems to be very convenient for task involving high accuracy and force-feedback control. We have noticed that the problems involved by the parallel design are dual to those found for classical manipulator. We have proposed results for the direct kinematic problem and for the symbolic expression of the jacobian matrix. The working area of such manipulators has been shown together with a simple method to verify if a trajectory is compatible with the limitation of the links lengths. The dynamic problem has been addressed and methods for force-feedback control introduced. In the second part of this paper we will address the difficult problem of the singular configurations.

## REFERENCES

### PARALLEL MANIPULATOR

**FICHTER E.F.**

"A Stewart platform based manipulator: general theory and practical construction", The Int. J. of Robotics Research, Vol.5, n 2, Summer 1986, pp. 157-181

**HUNT K.H.**

"Structural kinematics of in Parallel Actuated Robot Arms", Trans. of the ASME, J. of Mechanisms, Transmissions, and Automation in design, Vol 105, december 1983, pp.705-712

**INOUE H., TSUSAKA Y., FUKUIZUMI T.**

"Parallel manipulator", 3th ISRR, Gouvieux, France, 7-11 Oct.1985

**Mac CALLION H., PHAM D.T.**

"The analysis of a six degree of freedom work station for mechanised assembly", 5th World Congress on Theory of Machines and Mechanisms, Montreal, July 1979

**MERLET J-P.**

"Contribution a la formalisation de la commande par retour d'efforts. Application à la commande des robots parallèles" P.h.D. Thesis, Univ. Paris VI, 18 juin 1986

**MOHAMED M.G., DUFFY J.**

"A direct determination of the instantaneous kinematics of fully parallel robot manipulators", Trans. of the ASME, J. of Mechanisms, Transmissions, and Automation in design, Vol 107, june 1985, pp.226-229

**MINSKY M.**

"Manipulator design vignettes" MIT AIL, Memo 267, October 1972

**POWELL I.L., MIERE B.A.**

"The kinematic analysis and simulation of the parallel topology manipulator" The Marconi Review, Vol. XLV, n. 226, 1982, pp. 121-138

**REBOULET C., ROBERT A.**

"Hybrid control of a manipulator with an active compliant wrist", 3th ISRR, Gouvieux, France, 7-11 Oct.1985, pp.76-80

**TANAKA**

"Parallel manipulator" (in japanese) JAACE Symposium on flexible Automation, Osaka, Japan, 28-30 May 1985

**YANG D.C.H., LEE T.W.**

"Feasibility study of a platform type of robotic manipulator from a kinematic viewpoint", Trans. of the ASME, J. of Mechanisms, Transmissions, and Automation in design, Vol 106, june 1984, pp.191-198

**ZAMANOV V.B., SOTIROV Z.M.**

"Structures and kinematics of parallel topology manipulating systems", Int. Symp. on Design and Synthesis, July 11-13 1984, Tokyo, pp.453-458

### FLIGHT SIMULATOR

**STEWART D.**

"A platform with 6 degrees of freedom" Proc. of the institution of mechanical engineers 1965-66, Vol 180, part 1, number 15, pp.371-386

**WATSON P.** "Flight simulators-the grand illusion" Electron. Aust., Vol 46, num.4, pp.12-17

## **STRUCTURAL TOPOLOGY**

**CAUCHY A.**

"Deuxième mémoire sur les polygones et les polyèdres" Journal de l'Ecole Polytechnique, XVIème cahier, 1813

## **MACSYMA**

"MACSYMA Reference Manual", Version 10, December 1983

**HUSSAIN M.A., NOBLE B.**

"Application of symbolic computation to the analysis of mechanical systems, including robot arms", in Computer Aided Analysis and Optimization of Mechanical System Dynamics, E.J. Haug, NATO ASI Series F, Vol 9

```

/* INVERSE OF THE JACOBIAN MATRIX */

/* rotation matrix Me p=psi,t=teta,h=phi */

rot:matrix([cos(p)*cos(h)-sin(p)*cos(t)*sin(h), -cos(p)*sin(h)-sin(p)*
cos(t)*cos(h),sin(p)*sin(t)], [sin(p)*cos(h)+cos(p)*cos(t)*sin(h), -sin(p)*
sin(h)+cos(p)*cos(t)*cos(h), -cos(p)*sin(t)], [sin(t)*sin(h),sin(t)*cos(h),
cos(t)])$

(c4) /* relative coordinates of the articulation point on the mobile
plate */

blr:matrix([x1],[y1],[z1])$
(c5) b2r:matrix([x2],[y2],[z2])$
(c6) b3r:matrix([x3],[y3],[z3])$
(c7) b4r:matrix([x4],[y4],[z4])$
(c8) b5r:matrix([x5],[y5],[z5])$
(c9) b6r:matrix([x6],[y6],[z6])$

(c10) /* absolute coordinates of the articulation point on the fixed
plate */

a1:matrix([xa1],[ya1],[za1])$
(c11) a2:matrix([xa2],[ya2],[za2])$
(c12) a3:matrix([xa3],[ya3],[za3])$
(c13) a4:matrix([xa4],[ya4],[za4])$
(c14) a5:matrix([xa5],[ya5],[za5])$
(c15) a6:matrix([xa6],[ya6],[za6])$
(c16) ob1:rot.blr$
(c17) ob2:rot.b2r$
(c18) ob3:rot.b3r$
(c19) ob4:rot.b4r$
(c20) ob5:rot.b5r$
(c21) ob6:rot.b6r$

/* INVERSE OF THE JACOBIAN MATRIX , FIRST ROW ,ROW J IS
OBTAINED BY CHANGING THE INDICE 1 IN J */

invjac1:matrix([ xu[1],yu[1],zu[1],xu[1]*diff(ob1[1],p)+yu[1]*diff(ob1[2],p)+zu[1]
bl[1],t)+yu[1]*diff(ob1[2],t)+zu[1]*diff(ob1[3],t),xu[1]*diff(ob1[1],h)+yu[1]*di
,h)]);
(d22) matrix([xu , yu , zu , [yu (sin(p) sin(t) z1
1 1 1 1
+ (- cos(h) sin(p) cos(t) - sin(h) cos(p)) y1
+ (cos(h) cos(p) - sin(h) sin(p) cos(t)) x1)
+ xu (cos(p) sin(t) z1 + (sin(h) sin(p) - cos(h) cos(p) cos(t)) y1
1
+ (- sin(h) cos(p) cos(t) - cos(h) sin(p)) x1]],
[zu (- sin(t) z1 + cos(h) cos(t) y1 + sin(h) cos(t) x1)
1
+ xu (sin(p) cos(t) z1 + cos(h) sin(p) sin(t) y1 + sin(h) sin(p) sin(t) x1)
1
+ yu (- cos(p) cos(t) z1 - cos(h) cos(p) sin(t) y1
1
- sin(h) cos(p) sin(t) x1)], [zu (cos(h) sin(t) x1 - sin(h) sin(t) y1)
1

```

Jan 20 23:23 1987

APPENDIX 1 Page 2

```
+ xu1 ((sin(h) sin(p) cos(t) - cos(h) cos(p)) y1
+ (- cos(h) sin(p) cos(t) - sin(h) cos(p)) x1)
+ yu1 ((- sin(h) cos(p) cos(t) - cos(h) sin(p)) y1
+ (cos(h) cos(p) cos(t) - sin(h) sin(p)) x1)])
```

```

/* solve the linear system to get x0,y0,z0 position of the
center of the mobile according to the orientation
parameters and link's lengths ri (with rij= ri**2-rj**2)
xi,yi,zi: position of the articulation points on the mobile
xai,yai,zai: position of the articulation points on the base
bi:elements of the rotation matrix
*/

```

# LINEAR SYSTEM

```

eq1: -4*b7*x2*z0-4*b4*x2*y0+4*x0*(xa2-b1*x2)=r12-4*(b4*x2*ya1+b2*xa2*y1)$

```

```

eq2: 4*b7*x4*z0+4*b4*x4*y0+4*x0*(b1*x4-xa4)=r45+4*(b4*x4*ya4+b2*xa4*y4)$

```

```

eq3: 2*z0*(b8*(y3-y4)+b7*(x4-x3))+2*y0*(ya4-ya3+b5*(y3-y4)+b4*(x4-x3))
+2*x0*(b2*(y3-y4)+xa3-xa4+b1*(x4-x3))=r65-2*ya4*(b5*y4-b4*x4)-2*ya3*(
b4*x3-b5*y3)-2*b2*(xa3*y3-xa4*y4)-2*b1*(x4*xa4-x3*xa3)$

```

```

(c6) linsolve([eq1,eq2,eq3],[x0,y0,z0]),globalsolve:true$

```

## VALUE OF X0

```

(d8) - (4 b4 x2 x4 ya4 - 4 b4 x2 x4 ya1 + 4 b2 x2 xa4 y4 - 4 b2 x4 xa2 y1
+ r12 x4 + r45 x2)/(4 x2 xa4 - 4 x4 xa2)

```

## VALUE OF Z0

```

(d11) - (b4 (b1 (- 4 x2 x4 ya42 + (4 x2 x4 ya3 + 4 x2 x4 ya1) ya42
- 4 x2 x4 ya1 ya3 - 4 x2 x4 xa42 + (4 x2 x3 xa3 + 4 x4 xa2) xa42
- 4 x3 x4 xa2 xa3) + 4 x4 xa2 ya42 + (- 4 x4 xa2 ya3 - 4 x2 xa4 ya1) ya42
+ 4 x2 xa4 ya1 ya3 + b2 (- 4 x2 xa4 y42 + (4 x2 xa4 y3 + 4 x4 xa2 y1) y42
- 4 x4 xa2 y1 y3) + b2 (((4 x2 xa3 - 4 x3 xa2) xa4 - r12 x4 - r45 x2) y42
+ (- 4 x2 xa3 xa4 + 4 x4 xa2 xa3 + r12 x4 + r45 x2) y32
+ (4 x3 xa2 xa4 - 4 x4 xa2 xa3) y1) + ((2 r65 - r45) x2 - r12 x3) xa42
+ (r12 x4 + r45 x2) xa3 + ((r45 - 2 r65) x4 - r45 x3) xa2)
+ b5 (b4 (b1 ((4 x2 x4 y4 - 4 x2 x4 y3) ya4 + (4 x2 x4 y3 - 4 x2 x4 y4) ya1)
+ (4 x4 xa2 y3 - 4 x2 xa4 y4) ya4 + (4 x2 xa4 - 4 x4 xa2) y3 ya3
+ (4 x2 xa4 y4 - 4 x2 xa4 y3) ya1) + b22
(b1 (4 x2 xa4 y42 + (- 4 x2 xa4 y3 - 4 x4 xa2 y1) y4 + 4 x4 xa2 y1 y3)
- 4 xa2 xa4 y42 + (4 xa2 xa4 y3 + 4 xa2 xa4 y1) y4 - 4 xa2 xa4 y1 y3)
+ b1 ((r12 x4 + r45 x2) y4 + (- r12 x4 - r45 x2) y3)
+ (- r12 xa4 - r45 xa2) y4 + (r12 xa4 + r45 xa2) y3)
+ b2 (b1 ((4 x4 xa2 y1 - 4 x2 xa4 y4) ya4 + (4 x2 xa4 y4 - 4 x4 xa2 y1) ya3)
+ (4 xa2 xa4 y4 - 4 xa2 xa4 y1) ya4 + (4 xa2 xa4 y1 - 4 xa2 xa4 y4) ya3)
+ b4 (b2 ((4 x2 x4 y3 - 4 x2 x4 y4) ya4 + (4 x2 x4 y4 - 4 x2 x4 y3) ya1)
+ (4 x2 x4 xa3 - 4 x3 x4 xa2) ya4 + (4 x3 x4 xa2 - 4 x2 x3 xa4) ya3
+ (4 x2 x3 xa4 - 4 x2 x4 xa3) ya1) + b1
((- r12 x4 - r45 x2) ya4 + (r12 x4 + r45 x2) ya3) + (r12 xa4 + r45 xa2) ya4
+ (- r12 xa4 - r45 xa2) ya3)/(b7 ((4 x2 xa4 - 4 x4 xa2) ya42
+ (4 x4 xa2 - 4 x2 xa4) ya3 + b5 ((4 x4 xa2 - 4 x2 xa4) y42
+ (4 x2 xa4 - 4 x4 xa2) y3)) + b4 b8 ((4 x2 xa4 - 4 x4 xa2) y42

```

+ (4 x4 xa2 - 4 x2 xa4) y3))

VALUE OF Y0

(d12) (b8 (b4 (b1 ((4 x2 x4 y4 - 4 x2 x4 y3) ya4  
+ (4 x2 x4 y3 - 4 x2 x4 y4) ya1) + (4 x4 xa2 y3 - 4 x4 xa2 y4) ya4  
+ (4 x2 xa4 y4 - 4 x2 xa4 y3) ya1) + b2  
<sup>2</sup>  
(b1 (4 x2 xa4 y4<sup>2</sup> + (- 4 x2 xa4 y3 - 4 x4 xa2 y1) y4 + 4 x4 xa2 y1 y3)  
<sup>2</sup>  
- 4 xa2 xa4 y4 + (4 xa2 xa4 y3 + 4 xa2 xa4 y1) y4 - 4 xa2 xa4 y1 y3)  
+ b1 ((r12 x4 + r45 x2) y4 + (- r12 x4 - r45 x2) y3)  
+ (- r12 xa4 - r45 xa2) y4 + (r12 xa4 + r45 xa2) y3)  
+ b7 (b4 (b2 ((4 x2 x4 y3 - 4 x2 x4 y4) ya4 + (4 x2 x4 y4 - 4 x2 x4 y3) ya1)  
+ (4 x2 x4 xa3 - 4 x3 x4 xa2) ya4 + (4 x3 x4 xa2 - 4 x2 x3 xa4) ya3  
+ (4 x2 x3 xa4 - 4 x2 x4 xa3) ya1) + b5  
((4 x4 xa2 - 4 x2 xa4) y4 ya4 + (4 x2 xa4 - 4 x4 xa2) y3 ya3)  
<sup>2</sup> <sup>2</sup>  
+ b2 (- 4 x2 xa4 y4 + (4 x2 xa4 y3 + 4 x4 xa2 y1) y4 - 4 x4 xa2 y1 y3)  
+ b2 (((4 x2 xa3 - 4 x3 xa2) xa4 - r12 x4 - r45 x2) y4  
+ (- 4 x2 xa3 xa4 + 4 x4 xa2 xa3 + r12 x4 + r45 x2) y3  
+ (4 x3 xa2 xa4 - 4 x4 xa2 xa3) y1) + b1  
<sup>2</sup> <sup>2</sup>  
(- 4 x2 x4 xa4 + (4 x2 x3 xa3 + 4 x4 xa2) xa4 - 4 x3 x4 xa2 xa3)  
+ ((2 r65 - r45) x2 - r12 x3) xa4 + (r12 x4 + r45 x2) xa3  
+ ((r45 - 2 r65) x4 - r45 x3) xa2))/b7  
((4 x2 xa4 - 4 x4 xa2) ya4 + (4 x4 xa2 - 4 x2 xa4) ya3  
+ b5 ((4 x4 xa2 - 4 x2 xa4) y4 + (4 x2 xa4 - 4 x4 xa2) y3))  
+ b4 b8 ((4 x2 xa4 - 4 x4 xa2) y4 + (4 x4 xa2 - 4 x2 xa4) y3))



```

/*
calculate the Euler's angle of the mobile if we have
the unit vector of link 1,2,3 and the position of the
mobile x0,y0,z0. The rotation matrix M is of the form
      [ b1  b2  b3 ]
M=    [ b4  b5  b6 ] link's length ro[i],unit vector n[i]
      [ b7  b8  b9 ]
*/

eq1:b7*x1+b8*y1+b9*z1= -z0+za[1]+ro[1]*n1[3]$

(c4) eq2:b7*x2+b8*y2+b9*z2= -z0+za[2]+ro[2]*n2[3]$

(c5) eq3:b7*x3+b8*y3+b9*z3= -z0+za[3]+ro[3]*n3[3]$

(c6) linsolve([eq1,eq2,eq3],[b7,b8,b9]),globalsolve:true$

(c7) /*  VALUE OF B7
-----*/

b7;
(d7) - (z0 (y1 (z3 - z2) - y2 z3 + y3 z2) + ro1 n13 (y2 z3 - y3 z2)

+ za1 (y2 z3 - y3 z2) + y1 (- ro2 n23 z3 - za2 z3 + (za3 + n33 ro3) z2)

+ ((y2 - y3) z0 + ro2 n23 y3 + za2 y3 + (- za3 - n33 ro3) y2) z1)

/(x1 (y3 z2 - y2 z3) + y1 (x2 z3 - x3 z2) + (x3 y2 - x2 y3) z1)

(c8) /*-----*/

/* VALUE OF B8
-----*/

b8;
(d8) (z0 (x1 (z3 - z2) - x2 z3 + x3 z2) + ro1 n13 (x2 z3 - x3 z2)

+ za1 (x2 z3 - x3 z2) + x1 (- ro2 n23 z3 - za2 z3 + (za3 + n33 ro3) z2)

+ ((x2 - x3) z0 + ro2 n23 x3 + za2 x3 + (- za3 - n33 ro3) x2) z1)

/(x1 (y3 z2 - y2 z3) + y1 (x2 z3 - x3 z2) + (x3 y2 - x2 y3) z1)

(c9) /*-----*/

/* VALUE OF B9
-----*/

b9;

```

```

(d9) - ((x1 (y3 - y2) - x2 y3 + x3 y2 + (x2 - x3) y1) z0
+ ro1 n13 (x2 y3 - x3 y2) + za1 (x2 y3 - x3 y2)
+ x1 (- ro2 n23 y3 - za2 y3 + (za3 + n33 ro3) y2)
+ (ro2 n23 x3 + za2 x3 + (- za3 - n33 ro3) x2) y1)
/(x1 (y3 z2 - y2 z3) + y1 (x2 z3 - x3 z2) + (x3 y2 - x2 y3) z1)
(c10) /*-----*/
/* to get psi we must have b3 wich is the same as b9
if we change the subscript 3 in 1 */
teta=acos(b9) phi= asin(b7/sin(teta))
psi=asin(b3/sin(teta))

```

```
/* FORMULATION OF THE JACOBIAN MATRIX FOR A SSM*/
```

```
/* MATRIX J11 */
```

```
/* first row */
```

```
(d23)      - -----
              r1 x4
            2 x2 xa4 - 2 x4 xa2
              r2 x4
(d25)      - -----
            2 x2 xa4 - 2 x4 xa2
              0
              r4 x2
(d27)      - -----
            2 x2 xa4 - 2 x4 xa2
              r5 x2
(d29)      - -----
            2 x2 xa4 - 2 x4 xa2
              0
(d31)
```

```
/* 2nd row */
```

```
(d23) - r1 (b8 xa4 (y4 - y3) + b6 x4 (y4 - y3)
+ b7 ( x3 xa4 - x4 xa3))
/(2 ( x2 xa4 - x4 xa2) (b7 (ya4 - ya3) + b3 (y4 - y3))
(c24) /*2eme */
(d25) r2 (b8 xa4 (y4 - y3) + b6 x4 (y4 - y3) + b7 ( x3 xa4 - x4 xa3))
/(2 ( x2 xa4 - x4 xa2) (b7 (ya4 - ya3) + b3 (y4 - y3))
/3th element /      = 0
(c26) /*4eme*/
(d27) - r4 (b8 xa2 (y4 - y3) + b6 x2 (y4 - y3)
+ b7 ( x2 xa4 - x2 xa3) + b7 (x3 - x4) xa2)
/(2 ( x2 xa4 - x4 xa2) (b7 (ya4 - ya3) + b3 (y4 - y3))
(c28) /*5eme */
(d29) r5 (b8 xa2 (y4 - y3) + b6 x2 (y4 - y3)
- b7 ( x2 xa4 - x2 xa3) + b7 (x4 + x3) xa2)
/(2 ( x2 xa4 - x4 xa2) (b7 (ya4 - ya3) + b3 (y4 - y3))
(c30) /*6eme */
              b7 r6
(d31)      - -----
            b7 (ya4 - ya3) + b3 (y4 - y3)
```

```
/* 3th row */
```

```
(c22) /*1er element */
```

```
(d23) - ((r1 xa4 - b1 r1 x4) ya4 + (b1 r1 x4 - r1 xa4) ya3
+ (b9 r1 x4 - b5 r1 xa4) y4
+ (b5 r1 xa4 + (- b9) r1 x4) y3 - b4 r1 x3 xa4 + b4 r1 x4 xa3)
/((2 b7 x2 xa4 - 2 b7 x4 xa2) ya4 + (2 b7 x4 xa2 - 2 b7 x2 xa4) ya3
+ ((2 b4 b8 - 2 b5 b7) x2 xa4 + (2 b5 b7 - 2 b4 b8) x4 xa2) y4
+ ((2 b5 b7 - 2 b4 b8) x2 xa4 + (2 b4 b8 - 2 b5 b7) x4 xa2) y3)
(c24) /*2eme */
(d25) ((r2 xa4 - b1 r2 x4) ya4 + (b1 r2 x4 - r2 xa4) ya3
+ (b9 r2 x4 - b5 r2 xa4) y4
+ (b5 r2 xa4 + (- b9) r2 x4) y3 - b4 r2 x3 xa4 + b4 r2 x4 xa3)
/((2 b7 x2 xa4 - 2 b7 x4 xa2) ya4 + (2 b7 x4 xa2 - 2 b7 x2 xa4) ya3
+ ((2 b4 b8 - 2 b5 b7) x2 xa4 + (2 b5 b7 - 2 b4 b8) x4 xa2) y4
```

```

+ ((2 b5 b7 - 2 b4 b8) x2 xa4 + (2 b4 b8 - 2 b5 b7) x4 xa2) y3)
/* 3th element */ 0
(c26) /*4eme*/
(d27) - ((r4 xa2 - b1 r4 x2) ya4 + (b1 r4 x2 - r4 xa2) ya3
+ (b9 r4 x2 - b5 r4 xa2) y4
+ (b5 r4 xa2 + (- b9) r4 x2) y3 - b4 r4 x2 xa4 + b4 r4 x2 xa3
+ (b4 r4 x4 - b4 r4 x3) xa2)/((2 b7 x2 xa4 - 2 b7 x4 xa2) ya4
+ (2 b7 x4 xa2 - 2 b7 x2 xa4) ya3 + ((2 b4 b8 - 2 b5 b7) x2 xa4
+ (2 b5 b7 - 2 b4 b8) x4 xa2) y4 + ((2 b5 b7 - 2 b4 b8) x2 xa4
+ (2 b4 b8 - 2 b5 b7) x4 xa2) y3)
(c28) /*5eme */
(d29) ((r5 xa2 - b1 r5 x2) ya4 + (b1 r5 x2 - r5 xa2) ya3
+ (b9 r5 x2 - b5 r5 xa2) y4
+ (b5 r5 xa2 + (- b9) r5 x2) y3 + b4 r5 x2 xa4 + b4 r5 x2 xa3
+ (- b4 r5 x4 - b4 r5 x3) xa2)/((2 b7 x2 xa4 - 2 b7 x4 xa2) ya4
+ (2 b7 x4 xa2 - 2 b7 x2 xa4) ya3 + ((2 b4 b8 - 2 b5 b7) x2 xa4
+ (2 b5 b7 - 2 b4 b8) x4 xa2) y4 + ((2 b5 b7 - 2 b4 b8) x2 xa4
+ (2 b4 b8 - 2 b5 b7) x4 xa2) y3)
(c30) /*6eme */

(d31) - -----
          b4 r6
          b7 ya4 - b7 ya3 + (b4 b8 - b5 b7) y4 + (b5 b7 - b4 b8) y3

          -1
/* MATRIX (M+N1j12) */

/* COMMON DENOMINATOR */

(((z0 + b8 y3 + b7 x3) --- + b8 (x3 z0 + b7 (x3 - y3)) - b7 y3 z0
          dh
+ b6 (y3 ya3 - b5 y32 - y0 y3 - b4 x3 y3) + b5 (- x3 ya3 + x3 y0 + b4 x32)
          dy0
+ --- (- ya3 + b5 y3 + y0 + b4 x3) + b2 (b1 (x32 - y32) - x3 xa3 + x0 x3)
          dh
          dx0
+ --- (b2 y3 - xa3 + b1 x3 + x0) + b1 (xa3 - x0) y3 + b82 x3 y3 - b72 x3 y3
          dh
+ b52 x3 y3 + b22 x3 y3 - b12 x3 y3) (((z0 + b8 y1 - b7 x2) ---
          dz0
          dp
+ b2 (y0 y1 - y1 ya1) + b1 (x2 ya1 - x2 y0) + --- (- ya1 + b5 y1 + y0 - b4 x2)
          dy0
          dp
+ --- (b2 y1 + xa2 - b1 x2 + x0) + b5 (- xa2 - x0) y1 + b4 (x2 xa2 + x0 x2))
          dp
((z0 + b8 y4 + b7 x4) --- + (--- y4 + --- x4) z0 + (- --- y4 - --- x4) ya4
          dz0      db8      db7      db5      db4
          dt      dt      dt      dt      dt
+ --- (- ya4 + b5 y4 + y0 + b4 x4) + b8 (--- y42 + --- x4 y4)
          dy0      db8      db7
          dt      dt      dt
+ b5 (--- y42 + --- x4 y4) + b2 (--- y42 + --- x4 y4)
          db5      db4      db2      db1
          dt      dt      dt      dt

```

$$\begin{aligned}
& + b7 \left( \frac{db8}{dt} x4 y4 + \frac{db7}{dt} x4^2 \right) + b4 \left( \frac{db5}{dt} x4 y4 + \frac{db4}{dt} x4^2 \right) \\
& + b1 \left( \frac{db2}{dt} x4 y4 + \frac{db1}{dt} x4^2 \right) + \frac{dx0}{dt} (b2 y4 - xa4 + b1 x4 + x0) \\
& + \left( \frac{db5}{dt} y0 - \frac{db2}{dt} xa4 + \frac{db2}{dt} x0 \right) y4 + \frac{db4}{dt} x4 y0 - \frac{db1}{dt} x4 xa4 + \frac{db1}{dt} x0 x4 \\
& - \left( (z0 + b8 y4 + b7 x4) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya4) + b1 (x2 y0 - x2 ya4) \right. \\
& \left. + \frac{dy0}{dp} (- ya4 + b5 y4 + y0 + b4 x4) + b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1) \right. \\
& \left. + b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) + \frac{dx0}{dp} (b2 y4 - xa4 + b1 x4 + x0) \right) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} + \left( \frac{db8}{dt} y1 - \frac{db7}{dt} x2 \right) z0 + \left( \frac{db4}{dt} x2 - \frac{db5}{dt} y1 \right) ya1 \\
& + \frac{dy0}{dt} (- ya1 + b5 y1 + y0 - b4 x2) + b8 \left( \frac{db8}{dt} y1^2 - \frac{db7}{dt} x2 y1 \right) \\
& + b5 \left( \frac{db5}{dt} y1^2 - \frac{db4}{dt} x2 y1 \right) + b2 \left( \frac{db2}{dt} y1^2 - \frac{db1}{dt} x2 y1 \right) \\
& + b7 \left( \frac{db7}{dt} x2^2 - \frac{db8}{dt} x2 y1 \right) + b4 \left( \frac{db4}{dt} x2^2 - \frac{db5}{dt} x2 y1 \right) \\
& + b1 \left( \frac{db1}{dt} x2^2 - \frac{db2}{dt} x2 y1 \right) + \frac{dx0}{dt} (b2 y1 + xa2 - b1 x2 + x0) \\
& + \left( \frac{db5}{dt} y0 + \frac{db2}{dt} xa2 + \frac{db2}{dt} x0 \right) y1 - \frac{db4}{dt} x2 y0 - \frac{db1}{dt} x2 xa2 - \frac{db1}{dt} x0 x2) \\
& + \left( (z0 + b8 y3 + b7 x3) \frac{dz0}{dp} + \frac{dy0}{dp} (- ya3 + b5 y3 + y0 + b4 x3) \right. \\
& \left. + \frac{dx0}{dp} (b2 y3 - xa3 + b1 x3 + x0) + b8 \right) \\
& \left( (z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) - b7 y1 z0 \right. \\
& + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) \\
& + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) \\
& + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) + b8 x2 y4 + b5 x2 y4 + b2 x2 y4 \\
& + b1 (xa4 - x0) y1 - b7 x4 y1 - b1 x4 y1) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} + \left( \frac{db8}{dt} y1 - \frac{db7}{dt} x2 \right) z0 + \left( \frac{db4}{dt} x2 - \frac{db5}{dt} y1 \right) ya1
\end{aligned}$$

$$\begin{aligned}
& + \frac{dy_0}{dt} (-ya_1 + b_5 y_1 + y_0 - b_4 x_2) + b_8 \left( \frac{db_8}{dt} y_1^2 - \frac{db_7}{dt} x_2 y_1 \right) \\
& + b_5 \left( \frac{db_5}{dt} y_1^2 - \frac{db_4}{dt} x_2 y_1 \right) + b_2 \left( \frac{db_2}{dt} y_1^2 - \frac{db_1}{dt} x_2 y_1 \right) \\
& + b_7 \left( \frac{db_7}{dt} x_2^2 - \frac{db_8}{dt} x_2 y_1 \right) + b_4 \left( \frac{db_4}{dt} x_2^2 - \frac{db_5}{dt} x_2 y_1 \right) \\
& + b_1 \left( \frac{db_1}{dt} x_2^2 - \frac{db_2}{dt} x_2 y_1 \right) + \frac{dx_0}{dt} (b_2 y_1 + xa_2 - b_1 x_2 + x_0) \\
& + \left( \frac{db_5}{dt} y_0 + \frac{db_2}{dt} xa_2 + \frac{db_2}{dt} x_0 \right) y_1 - \frac{db_4}{dt} x_2 y_0 - \frac{db_1}{dt} x_2 xa_2 - \frac{db_1}{dt} x_0 x_2 \\
& - ((z_0 + b_8 y_1 - b_7 x_2) \frac{dz_0}{dh} + b_8 (b_7 (x_2^2 - y_1^2) - x_2 z_0) - b_7 y_1 z_0 \\
& + b_6 (y_1 ya_1 - b_5 y_1^2 - y_0 y_1 + b_4 x_2 y_1) + b_5 (x_2 ya_1 - x_2 y_0 + b_4 x_2^2) \\
& + \frac{dy_0}{dh} (-ya_1 + b_5 y_1 + y_0 - b_4 x_2) + b_2 (b_1 (x_2^2 - y_1^2) - x_2 xa_2 - x_0 x_2) \\
& + \frac{dx_0}{dh} (b_2 y_1 + xa_2 - b_1 x_2 + x_0) + b_1 (-xa_2 - x_0) y_1 - b_8 x_2 y_1 + b_7 x_2 y_1 \\
& - b_5 x_2 y_1 - b_2 x_2 y_1 + b_1 x_2 y_1) ((z_0 + b_8 y_4 + b_7 x_4) \frac{dz_0}{dt} \\
& + \left( \frac{db_8}{dt} y_4 + \frac{db_7}{dt} x_4 \right) z_0 + \left( -\frac{db_5}{dt} y_4 - \frac{db_4}{dt} x_4 \right) ya_4 \\
& + \frac{dy_0}{dt} (-ya_4 + b_5 y_4 + y_0 + b_4 x_4) + b_8 \left( \frac{db_8}{dt} y_4^2 + \frac{db_7}{dt} x_4 y_4 \right) \\
& + b_5 \left( \frac{db_5}{dt} y_4^2 + \frac{db_4}{dt} x_4 y_4 \right) + b_2 \left( \frac{db_2}{dt} y_4^2 + \frac{db_1}{dt} x_4 y_4 \right) \\
& + b_7 \left( \frac{db_7}{dt} x_4 y_4 + \frac{db_8}{dt} x_4^2 \right) + b_4 \left( \frac{db_4}{dt} x_4 y_4 + \frac{db_5}{dt} x_4^2 \right) \\
& + b_1 \left( \frac{db_1}{dt} x_4 y_4 + \frac{db_2}{dt} x_4^2 \right) + \frac{dx_0}{dt} (b_2 y_4 - xa_4 + b_1 x_4 + x_0) \\
& + \left( \frac{db_5}{dt} y_0 - \frac{db_2}{dt} xa_4 + \frac{db_2}{dt} x_0 \right) y_4 + \frac{db_4}{dt} x_4 y_0 - \frac{db_1}{dt} x_4 xa_4 + \frac{db_1}{dt} x_0 x_4) \\
& + ((z_0 + b_8 y_1 - b_7 x_2) \frac{dz_0}{dh} + b_8 (b_7 (x_2^2 - y_1^2) - x_2 z_0) - b_7 y_1 z_0 \\
& + b_6 (y_1 ya_1 - b_5 y_1^2 - y_0 y_1 + b_4 x_2 y_1) + b_5 (x_2 ya_1 - x_2 y_0 + b_4 x_2^2) \\
& + \frac{dy_0}{dh} (-ya_1 + b_5 y_1 + y_0 - b_4 x_2) + b_2 (b_1 (x_2^2 - y_1^2) - x_2 xa_2 - x_0 x_2)
\end{aligned}$$

$$\begin{aligned}
& + \frac{dx0}{dh} (b2 y1 + xa2 - b1 x2 + x0) + b1 (- xa2 - x0) y1 - b8^2 x2 y1 + b7^2 x2 y1 \\
& - b5^2 x2 y1 - b2^2 x2 y1 + b1^2 x2 y1) ((z0 + b8 y4 + b7 x4) \frac{dz0}{dp} \\
& + b2 (y0 y1 - y1 ya4) + b1 (x2 y0 - x2 ya4) + \frac{dy0}{dp} (- ya4 + b5 y4 + y0 + b4 x4) \\
& + b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1) \\
& + b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) + \frac{dx0}{dp} (b2 y4 - xa4 + b1 x4 + x0)) \\
& - ((z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) - b7 y1 z0 \\
& + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) \\
& + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) \\
& + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) + b8^2 x2 y4 + b5^2 x2 y4 + b2^2 x2 y4 \\
& + b1 (xa4 - x0) y1 - b7^2 x4 y1 - b1^2 x4 y1) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya1) + b1 (x2 ya1 - x2 y0) \\
& + \frac{dy0}{dp} (- ya1 + b5 y1 + y0 - b4 x2) + \frac{dx0}{dp} (b2 y1 + xa2 - b1 x2 + x0) \\
& + b5 (- xa2 - x0) y1 + b4 (x2 xa2 + x0 x2))) \\
& ((z0 + b8 y3 + b7 x3) \frac{dz0}{dt} + \frac{dy0}{dt} (- ya3 + b5 y3 + y0 + b4 x3) \\
& + \frac{dx0}{dt} (b2 y3 - xa3 + b1 x3 + x0) + b8)) ]
\end{aligned}$$

/\* 1ST ROW, 1ST ELEMENT \*/

$$\begin{aligned}
(d101) & [- ((z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) \\
& - b7 y1 z0 + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) \\
& + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) \\
& + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) \\
& + b8^2 x2 y4 + b5^2 x2 y4 + b2^2 x2 y4 + b1 (xa4 - x0) y1 - b7^2 x4 y1 \\
& - b1^2 x4 y1) ((z0 + b8 y3 + b7 x3) \frac{dz0}{dt} + \frac{dy0}{dt} (- ya3 + b5 y3 + y0 + b4 x3)
\end{aligned}$$

$$\begin{aligned}
& + \frac{dx0}{dt} (b2 y3 - xa3 + b1 x3 + x0) + b8) \\
& - ((z0 + b8 y3 + b7 x3) \frac{dz0}{dh} + b8 (x3 z0 + b7 (x3^2 - y3^2)) - b7 y3 z0 \\
& + b6 \frac{dy0}{dh} (y3 ya3 - b5 y3^2 - y0 y3 - b4 x3 y3) + b5 (- x3 ya3 + x3 y0 + b4 x3^2) \\
& + \frac{dh}{dh} (- ya3 + b5 y3 + y0 + b4 x3) + b2 (b1 (x3^2 - y3^2) - x3 xa3 + x0 x3) \\
& + \frac{dx0}{dh} (b2 y3 - xa3 + b1 x3 + x0) + b1 (xa3 - x0) y3 + b8^2 x3 y3 - b7^2 x3 y3 \\
& + b5^2 x3 y3 + b2^2 x3 y3 - b1^2 x3 y3) ((z0 + b8 y4 + b7 x4) \frac{dz0}{dt} \\
& + \frac{db8}{dt} y4 + \frac{db7}{dt} x4) z0 + (- \frac{db5}{dt} y4 - \frac{db4}{dt} x4) ya4 \\
& + \frac{dy0}{dt} (- ya4 + b5 y4 + y0 + b4 x4) + b8 \frac{db8}{dt} y4^2 + \frac{db7}{dt} x4 y4) \\
& + b5 \frac{db5}{dt} y4^2 + \frac{db4}{dt} x4 y4) + b2 \frac{db2}{dt} y4^2 + \frac{db1}{dt} x4 y4) \\
& + b7 \frac{db8}{dt} x4 y4 + \frac{db7}{dt} x4^2) + b4 \frac{db5}{dt} x4 y4 + \frac{db4}{dt} x4^2) \\
& + b1 \frac{db2}{dt} x4 y4 + \frac{db1}{dt} x4^2) + \frac{dx0}{dt} (b2 y4 - xa4 + b1 x4 + x0) \\
& + (- \frac{db5}{dt} y0 - \frac{db2}{dt} xa4 + \frac{db2}{dt} x0) y4 + \frac{db4}{dt} x4 y0 - \frac{db1}{dt} x4 xa4 + \frac{db1}{dt} x0 x4)) \\
& /denom
\end{aligned}$$

/\* 1ST ROW, 2ND ELEMENT \*/

$$\begin{aligned}
(d102) \quad & [- ((z0 + b8 y3 + b7 x3) \frac{dz0}{dh} + b8 (x3 z0 + b7 (x3^2 - y3^2)) - b7 y3 z0 \\
& + b6 \frac{dy0}{dh} (y3 ya3 - b5 y3^2 - y0 y3 - b4 x3 y3) + b5 (- x3 ya3 + x3 y0 + b4 x3^2) \\
& + \frac{dh}{dh} (- ya3 + b5 y3 + y0 + b4 x3) + b2 (b1 (x3^2 - y3^2) - x3 xa3 + x0 x3) \\
& + \frac{dx0}{dh} (b2 y3 - xa3 + b1 x3 + x0) + b1 (xa3 - x0) y3 + b8^2 x3 y3 - b7^2 x3 y3 \\
& + b5^2 x3 y3 + b2^2 x3 y3 - b1^2 x3 y3) ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} \\
& + \frac{db8}{dt} y1 - \frac{db7}{dt} x2) z0 + (- \frac{db4}{dt} x2 - \frac{db5}{dt} y1) ya1 \\
& \frac{db8}{dt} y1^2 + \frac{db7}{dt} x2 y1)
\end{aligned}$$



```

+ --- (- ya1 + b5 y1 + y0 - b4 x2) + b8 (--- y1 - --- x2 y1)
dt dt dt dt
+ b5 (--- y1 - --- x2 y1) + b2 (--- y1 - --- x2 y1)
dt dt dt dt
+ b7 (--- x2 - --- x2 y1) + b4 (--- x2 - --- x2 y1)
dt dt dt dt
+ b1 (--- x2 - --- x2 y1) + --- (b2 y1 + xa2 - b1 x2 + x0)
dt dt dt dt
+ (--- y0 + --- xa2 + --- x0) y1 - --- x2 y0 - --- x2 xa2 - --- x0 x2)
dt dt dt dt dt dt
- ((z0 + b8 y1 - b7 x2) --- + b8 (b7 (x2 - y1 ) - x2 z0) - b7 y1 z0
dh 2 2
+ b6 (y1 ya1 - b5 y1 - y0 y1 + b4 x2 y1) + b5 (x2 ya1 - x2 y0 + b4 x2 )
dy0 2 2
+ --- (- ya1 + b5 y1 + y0 - b4 x2) + b2 (b1 (x2 - y1 ) - x2 xa2 - x0 x2)
dh 2 2
+ --- (b2 y1 + xa2 - b1 x2 + x0) + b1 (- xa2 - x0) y1 - b8 x2 y1 + b7 x2 y1
dh 2 2
- b5 x2 y1 - b2 x2 y1 + b1 x2 y1) ((z0 + b8 y3 + b7 x3) ---
2 2 2 dz0
+ --- (- ya3 + b5 y3 + y0 + b4 x3) + --- (b2 y3 - xa3 + b1 x3 + x0) + b8))
dt dt dt
/denom
/* 1ST ROW 3TH ELEMENT */

```

```

(d103) [- (((z0 + b8 y1 - b7 x2) --- + b8 (b7 (x2 - y1 ) - x2 z0) - b7 y1 z0
dh 2 2
+ b6 (y1 ya1 - b5 y1 - y0 y1 + b4 x2 y1) + b5 (x2 ya1 - x2 y0 + b4 x2 )
dy0 2 2
+ --- (- ya1 + b5 y1 + y0 - b4 x2) + b2 (b1 (x2 - y1 ) - x2 xa2 - x0 x2)
dh 2 2
+ --- (b2 y1 + xa2 - b1 x2 + x0) + b1 (- xa2 - x0) y1 - b8 x2 y1 + b7 x2 y1
dh 2 2
- b5 x2 y1 - b2 x2 y1 + b1 x2 y1) ((z0 + b8 y4 + b7 x4) ---
2 2 2 dz0
+ --- y4 + --- x4) z0 + (- --- y4 - --- x4) ya4
dt dt dt dt
+ --- (- ya4 + b5 y4 + y0 + b4 x4) + b8 (--- y4 + --- x4 y4)
dt dt dt 2
db5 2 db4 db2 2 db1

```

$$\begin{aligned}
& + b5 \left( \frac{---}{dt} y4 + \frac{---}{dt} x4 y4 \right) + b2 \left( \frac{---}{dt} y4 + \frac{---}{dt} x4 y4 \right) \\
& + b7 \left( \frac{---}{dt} x4 y4 + \frac{---}{dt} x4^2 \right) + b4 \left( \frac{---}{dt} x4 y4 + \frac{---}{dt} x4^2 \right) \\
& + b1 \left( \frac{---}{dt} x4 y4 + \frac{---}{dt} x4^2 \right) + \frac{dx0}{dt} (b2 y4 - xa4 + b1 x4 + x0) \\
& + \left( \frac{db5}{dt} y0 - \frac{db2}{dt} xa4 + \frac{db2}{dt} x0 \right) y4 + \frac{db4}{dt} x4 y0 - \frac{db1}{dt} x4 xa4 + \frac{db1}{dt} x0 x4 \\
& - ((z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) - b7 y1 z0 \\
& + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) \\
& + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) \\
& + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) + b8^2 x2 y4 + b5^2 x2 y4 + b2^2 x2 y4 \\
& + b1 (xa4 - x0) y1 - b7^2 x4 y1 - b1^2 x4 y1 \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} + \frac{db8}{dt} y1 - \frac{db7}{dt} x2) z0 + \left( \frac{db4}{dt} x2 - \frac{db5}{dt} y1 \right) ya1 \\
& + \frac{dy0}{dt} (- ya1 + b5 y1 + y0 - b4 x2) + b8 \left( \frac{db8}{dt} y1^2 - \frac{db7}{dt} x2 y1 \right) \\
& + b5 \left( \frac{db5}{dt} y1^2 - \frac{db4}{dt} x2 y1 \right) + b2 \left( \frac{db2}{dt} y1^2 - \frac{db1}{dt} x2 y1 \right) \\
& + b7 \left( \frac{db7}{dt} x2^2 - \frac{db8}{dt} x2 y1 \right) + b4 \left( \frac{db4}{dt} x2^2 - \frac{db5}{dt} x2 y1 \right) \\
& + b1 \left( \frac{db1}{dt} x2^2 - \frac{db2}{dt} x2 y1 \right) + \frac{dx0}{dt} (b2 y1 + xa2 - b1 x2 + x0) \\
& + \left( \frac{db5}{dt} y0 + \frac{db2}{dt} xa2 + \frac{db2}{dt} x0 \right) y1 - \frac{db4}{dt} x2 y0 - \frac{db1}{dt} x2 xa2 - \frac{db1}{dt} x0 x2) \\
& /denom \\
& /* 2ND ROW 1ST ELEMENT */ \\
& (d101) [(((z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) - b7 y1 z0 \\
& + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) \\
& + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) \\
& + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) + b8^2 x2 y4 + b5^2 x2 y4 + b2^2 x2 y4 \\
& 2 \qquad \qquad \qquad 2 \qquad \qquad \qquad 2
\end{aligned}$$

```

+ b1 (xa4 - x0) y1 - b7 x4 y1 - b1 x4 y1)
((z0 + b8 y3 + b7 x3)  $\frac{dz0}{dp}$  +  $\frac{dy0}{dp}$  (- ya3 + b5 y3 + y0 + b4 x3)
 $\frac{dx0}{dp}$  (b2 y3 - xa3 + b1 x3 + x0) + b8)
- ((z0 + b8 y3 + b7 x3)  $\frac{dz0}{dh}$  + b8 (x3 z0 + b7 (x32 - y32)) - b7 y3 z0
+ b6 (y3 ya3 - b5 y32 - y0 y3 - b4 x3 y3) + b5 (- x3 ya3 + x3 y0 + b4 x32)
 $\frac{dy0}{dh}$  (- ya3 + b5 y3 + y0 + b4 x3) + b2 (b1 (x32 - y32) - x3 xa3 + x0 x3)
 $\frac{dx0}{dh}$  (b2 y3 - xa3 + b1 x3 + x0) + b1 (xa3 - x0) y3 + b8 x3 y3 - b72 x3 y3
+ b52 x3 y3 + b22 x3 y3 - b12 x3 y3) ((z0 + b8 y4 + b7 x4)  $\frac{dz0}{dp}$ 
+ b2 (y0 y1 - y1 ya4) + b1 (x2 y0 - x2 ya4) +  $\frac{dy0}{dp}$  (- ya4 + b5 y4 + y0 + b4 x4)
+ b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1)
+ b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) +  $\frac{dx0}{dp}$  (b2 y4 - xa4 + b1 x4 + x0)))
/denom
/* 2ND ROW 2ND ELEMENT */
(d102) [(((z0 + b8 y3 + b7 x3)  $\frac{dz0}{dh}$  + b8 (x3 z0 + b7 (x32 - y32)) - b7 y3 z0
+ b6 (y3 ya3 - b5 y32 - y0 y3 - b4 x3 y3) + b5 (- x3 ya3 + x3 y0 + b4 x32)
 $\frac{dy0}{dh}$  (- ya3 + b5 y3 + y0 + b4 x3) + b2 (b1 (x32 - y32) - x3 xa3 + x0 x3)
 $\frac{dx0}{dh}$  (b2 y3 - xa3 + b1 x3 + x0) + b1 (xa3 - x0) y3 + b8 x3 y3 - b72 x3 y3
+ b52 x3 y3 + b22 x3 y3 - b12 x3 y3) ((z0 + b8 y1 - b7 x2)  $\frac{dz0}{dp}$ 
+ b2 (y0 y1 - y1 ya1) + b1 (x2 ya1 - x2 y0) +  $\frac{dy0}{dp}$  (- ya1 + b5 y1 + y0 - b4 x2)
 $\frac{dx0}{dp}$  (b2 y1 + xa2 - b1 x2 + x0) + b5 (- xa2 - x0) y1 + b4 (x2 xa2 + x0 x2))
- ((z0 + b8 y1 - b7 x2)  $\frac{dz0}{dh}$  + b8 (b7 (x22 - y12) - x2 z0) - b7 y1 z0
+ b6 (y1 ya1 - b5 y12 - y0 y1 + b4 x2 y1) + b5 (x2 ya1 - x2 y0 + b4 x22)

```

$$\begin{aligned}
& + \frac{dy0}{dh} (- ya1 + b5 y1 + y0 - b4 x2) + b2 (b1 (x2^2 - y1^2) - x2 xa2 - x0 x2) \\
& + \frac{dx0}{dh} (b2 y1 + xa2 - b1 x2 + x0) + b1 (- xa2 - x0) y1 - b8^2 x2 y1 + b7^2 x2 y1 \\
& - b5^2 x2 y1 - b2^2 x2 y1 + b1^2 x2 y1) ((z0 + b8 y3 + b7 x3) \frac{dz0}{dp} \\
& + \frac{dy0}{dp} (- ya3 + b5 y3 + y0 + b4 x3) + \frac{dx0}{dp} (b2 y3 - xa3 + b1 x3 + x0) + b8)) \\
& /denom
\end{aligned}$$

/\* 2ND ROW 3HD ELEMENT \*/

$$\begin{aligned}
(d103) & [(((z0 + b8 y1 - b7 x2) \frac{dz0}{dh} + b8 (b7 (x2^2 - y1^2) - x2 z0) - b7 y1 z0 \\
& + b6 (y1 ya1 - b5 y1^2 - y0 y1 + b4 x2 y1) + b5 (x2 ya1 - x2 y0 + b4 x2^2) \\
& + \frac{dy0}{dh} (- ya1 + b5 y1 + y0 - b4 x2) + b2 (b1 (x2^2 - y1^2) - x2 xa2 - x0 x2) \\
& + \frac{dx0}{dh} (b2 y1 + xa2 - b1 x2 + x0) + b1 (- xa2 - x0) y1 - b8^2 x2 y1 + b7^2 x2 y1 \\
& - b5^2 x2 y1 - b2^2 x2 y1 + b1^2 x2 y1) ((z0 + b8 y4 + b7 x4) \frac{dz0}{dp} \\
& + b2 (y0 y1 - y1 ya4) + b1 (x2 y0 - x2 ya4) + \frac{dy0}{dp} (- ya4 + b5 y4 + y0 + b4 x4) \\
& + b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1) \\
& + b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) + \frac{dx0}{dp} (b2 y4 - xa4 + b1 x4 + x0)) \\
& - ((z0 + b8 y4 + b7 x4) \frac{dz0}{dh} + b8 (x2 z0 + b7 (x2 x4 - y1 y4)) - b7 y1 z0 \\
& + b6 (y1 ya4 - b5 y1 y4 - y0 y1 - b4 x4 y1) + b5 (- x2 ya4 + x2 y0 + b4 x2 x4) \\
& + \frac{dy0}{dh} (- ya4 + b5 y4 + y0 + b4 x4) + b2 (b1 (x2 x4 - y1 y4) - x2 xa4 + x0 x2) \\
& + \frac{dx0}{dh} (b2 y4 - xa4 + b1 x4 + x0) + b8^2 x2 y4 + b5^2 x2 y4 + b2^2 x2 y4 \\
& + b1 (xa4 - x0) y1 - b7^2 x4 y1 - b1^2 x4 y1) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya1) + b1 (x2 ya1 - x2 y0) \\
& + \frac{dy0}{dp} (- ya1 + b5 y1 + y0 - b4 x2) + \frac{dx0}{dp} (b2 y1 + xa2 - b1 x2 + x0) \\
& + b5 (- xa2 - x0) y1 + b4 (x2 xa2 + x0 x2)))
\end{aligned}$$

/denom

/\* 3TH ROW 1ST ELEMENT \*/

$$\begin{aligned}
 (d101) \quad & \left[ - \left( (z0 + b8 y3 + b7 x3) \frac{dz0}{dp} + \frac{dy0}{dp} (-ya3 + b5 y3 + y0 + b4 x3) \right. \right. \\
 & + \frac{dx0}{dp} (b2 y3 - xa3 + b1 x3 + x0) + b8) \\
 & ((z0 + b8 y4 + b7 x4) \frac{dz0}{dt} + (\frac{db8}{dt} y4 + \frac{db7}{dt} x4) z0 + (-\frac{db5}{dt} y4 - \frac{db4}{dt} x4) ya4 \\
 & + \frac{dy0}{dt} (-ya4 + b5 y4 + y0 + b4 x4) + b8 (\frac{db8}{dt} y4^2 + \frac{db7}{dt} x4 y4) \\
 & + b5 (\frac{db5}{dt} y4^2 + \frac{db4}{dt} x4 y4) + b2 (\frac{db2}{dt} y4^2 + \frac{db1}{dt} x4 y4) \\
 & + b7 (\frac{db8}{dt} x4 y4 + \frac{db7}{dt} x4^2) + b4 (\frac{db5}{dt} x4 y4 + \frac{db4}{dt} x4^2) \\
 & + b1 (\frac{db2}{dt} x4 y4 + \frac{db1}{dt} x4^2) + \frac{dx0}{dt} (b2 y4 - xa4 + b1 x4 + x0) \\
 & + (\frac{db5}{dt} y0 - \frac{db2}{dt} xa4 + \frac{db2}{dt} x0) y4 + \frac{db4}{dt} x4 y0 - \frac{db1}{dt} x4 xa4 + \frac{db1}{dt} x0 x4) \\
 & - ((z0 + b8 y4 + b7 x4) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya4) + b1 (x2 y0 - x2 ya4) \\
 & + \frac{dy0}{dp} (-ya4 + b5 y4 + y0 + b4 x4) + b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1) \\
 & + b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) + \frac{dx0}{dp} (b2 y4 - xa4 + b1 x4 + x0)) \\
 & ((z0 + b8 y3 + b7 x3) \frac{dz0}{dt} + \frac{dy0}{dt} (-ya3 + b5 y3 + y0 + b4 x3) \\
 & + \frac{dx0}{dt} (b2 y3 - xa3 + b1 x3 + x0) + b8)) \\
 & /denom
 \end{aligned}$$

/\* 3TH ROW 2ND ELEMENT \*/

$$\begin{aligned}
 (d102) \quad & \left[ - \left( (z0 + b8 y1 - b7 x2) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya1) \right. \right. \\
 & + b1 (x2 ya1 - x2 y0) + \frac{dy0}{dp} (-ya1 + b5 y1 + y0 - b4 x2) \\
 & + \frac{dx0}{dp} (b2 y1 + xa2 - b1 x2 + x0) + b5 (-xa2 - x0) y1 + b4 (x2 xa2 + x0 x2)) \\
 & \frac{dz0}{dt} \quad \frac{dy0}{dt}
 \end{aligned}$$

$$\begin{aligned}
& ((z0 + b8 y3 + b7 x3) \frac{dx0}{dt} + \frac{dy0}{dt} (- ya3 + b5 y3 + y0 + b4 x3) \\
& + \frac{dx0}{dt} (b2 y3 - xa3 + b1 x3 + x0) + b8) \\
& - ((z0 + b8 y3 + b7 x3) \frac{dz0}{dp} + \frac{dy0}{dp} (- ya3 + b5 y3 + y0 + b4 x3) \\
& + \frac{dx0}{dp} (b2 y3 - xa3 + b1 x3 + x0) + b8) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} + (\frac{db8}{dt} y1 - \frac{db7}{dt} x2) z0 + (\frac{db4}{dt} x2 - \frac{db5}{dt} y1) ya1 \\
& + \frac{dy0}{dt} (- ya1 + b5 y1 + y0 - b4 x2) + b8 (\frac{db8}{dt} y1^2 - \frac{db7}{dt} x2 y1) \\
& + b5 (\frac{db5}{dt} y1^2 - \frac{db4}{dt} x2 y1) + b2 (\frac{db2}{dt} y1^2 - \frac{db1}{dt} x2 y1) \\
& + b7 (\frac{db7}{dt} x2^2 - \frac{db8}{dt} x2 y1) + b4 (\frac{db4}{dt} x2^2 - \frac{db5}{dt} x2 y1) \\
& + b1 (\frac{db1}{dt} x2^2 - \frac{db2}{dt} x2 y1) + \frac{dx0}{dt} (b2 y1 + xa2 - b1 x2 + x0) \\
& + (\frac{db5}{dt} y0 + \frac{db2}{dt} xa2 + \frac{db2}{dt} x0) y1 - \frac{db4}{dt} x2 y0 - \frac{db1}{dt} x2 xa2 - \frac{db1}{dt} x0 x2)) \\
& /denom
\end{aligned}$$

/\* 3TH ROW 3THD ELEMENT \*/

$$\begin{aligned}
& (d103) [- ((z0 + b8 y4 + b7 x4) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya4) \\
& + b1 (x2 y0 - x2 ya4) + \frac{dy0}{dp} (- ya4 + b5 y4 + y0 + b4 x4) \\
& + b5 (b1 (x2 y4 - x4 y1) + (xa4 - x0) y1) \\
& + b4 (b2 (x4 y1 - x2 y4) + x2 xa4 - x0 x2) + \frac{dx0}{dp} (b2 y4 - xa4 + b1 x4 + x0)) \\
& ((z0 + b8 y1 - b7 x2) \frac{dz0}{dt} + (\frac{db8}{dt} y1 - \frac{db7}{dt} x2) z0 + (\frac{db4}{dt} x2 - \frac{db5}{dt} y1) ya1 \\
& + \frac{dy0}{dt} (- ya1 + b5 y1 + y0 - b4 x2) + b8 (\frac{db8}{dt} y1^2 - \frac{db7}{dt} x2 y1) \\
& + b5 (\frac{db5}{dt} y1^2 - \frac{db4}{dt} x2 y1) + b2 (\frac{db2}{dt} y1^2 - \frac{db1}{dt} x2 y1) \\
& + b7 (\frac{db7}{dt} x2^2 - \frac{db8}{dt} x2 y1) + b4 (\frac{db4}{dt} x2^2 - \frac{db5}{dt} x2 y1) \\
& + b1 (\frac{db1}{dt} x2^2 - \frac{db2}{dt} x2 y1) + \frac{dx0}{dt} (b2 y1 + xa2 - b1 x2 + x0) \\
& + (\frac{db5}{dt} y0 + \frac{db2}{dt} xa2 + \frac{db2}{dt} x0) y1 - \frac{db4}{dt} x2 y0 - \frac{db1}{dt} x2 xa2 - \frac{db1}{dt} x0 x2))
\end{aligned}$$

$$\begin{aligned}
& + b1 \left( \frac{---}{dt} x2 - \frac{---}{dt} x2 y1 \right) + \frac{---}{dt} (b2 y1 + xa2 - b1 x2 + x0) \\
& + \left( \frac{db5}{dt} y0 + \frac{db2}{dt} xa2 + \frac{db2}{dt} x0 \right) y1 - \frac{db4}{dt} x2 y0 - \frac{db1}{dt} x2 xa2 - \frac{db1}{dt} x0 x2 \\
& - ((z0 + b8 y1 - b7 x2) \frac{dz0}{dp} + b2 (y0 y1 - y1 ya1) + b1 (x2 ya1 - x2 y0) \\
& + \frac{dy0}{dp} (- ya1 + b5 y1 + y0 - b4 x2) + \frac{dx0}{dp} (b2 y1 + xa2 - b1 x2 + x0) \\
& + b5 (- xa2 - x0) y1 + b4 (x2 xa2 + x0 x2)) \\
& ((z0 + b8 y4 + b7 x4) \frac{dz0}{dt} + \frac{db8}{dt} y4 + \frac{db7}{dt} x4) z0 + (- \frac{db5}{dt} y4 - \frac{db4}{dt} x4) ya4 \\
& + \frac{dy0}{dt} (- ya4 + b5 y4 + y0 + b4 x4) + b8 \left( \frac{db8}{dt} y4^2 + \frac{db7}{dt} x4 y4 \right) \\
& + b5 \left( \frac{db5}{dt} y4^2 + \frac{db4}{dt} x4 y4 \right) + b2 \left( \frac{db2}{dt} y4^2 + \frac{db1}{dt} x4 y4 \right) \\
& + b7 \left( \frac{db8}{dt} x4 y4 + \frac{db7}{dt} x4^2 \right) + b4 \left( \frac{db5}{dt} x4 y4 + \frac{db4}{dt} x4^2 \right) \\
& + b1 \left( \frac{db2}{dt} x4 y4 + \frac{db1}{dt} x4^2 \right) + \frac{dx0}{dt} (b2 y4 - xa4 + b1 x4 + x0) \\
& + \left( \frac{db5}{dt} y0 - \frac{db2}{dt} xa4 + \frac{db2}{dt} x0 \right) y4 + \frac{db4}{dt} x4 y0 - \frac{db1}{dt} x4 xa4 + \frac{db1}{dt} x0 x4) \\
& /denom
\end{aligned}$$

```
/* calculate the innacuracy on the position of the
center of the mobile according to the inaccuracy on
the measurement of the link's length when
psi=teta=phi=0
/*
```

```
/* linear system to be solved */
```

```
(c3) eq1:dz0*z0+(x0-xa1+x1)*dx0+(y0-ya1+y1)*dy0=r1*dr1$
(c4) eq2:dz0*z0+(x0-xa3+x3)*dx0+(y0-ya3+y3)*dy0=r3*dr3$
(c5) eq3:dz0*z0+(x0-xa4+x4)*dx0+(y0-ya4+y4)*dy0=r4*dr4$

(c6) linsolve([eq1,eq2,eq3],[dx0,dy0,dz0]),globalsolve:true$
```

INNACURACY ON X0

```
(c7) dx01:ratsimp(dx0,dr1,dr3,dr4);
(d7) (dr3 (- r3 ya4 + r3 ya1 + r3 y4 - r3 y1)
+ dr1 (r1 ya4 - r1 ya3 - r1 y4 + r1 y3)
+ dr4 (r4 ya3 - r4 ya1 - r4 y3 + r4 y1))
/((xa3 - xa1 - x3 + x1) ya4 + (- xa4 + xa1 + x4 - x1) ya3
+ (xa4 - xa3 - x4 + x3) ya1 + (- xa3 + xa1 + x3 - x1) y4
+ (xa4 - xa1 - x4 + x1) y3 + (- xa4 + xa3 + x4 - x3) y1)
```

INNACURACY ON Y0

```
(c8) dy01:ratsimp(dy0,dr1,dr3,dr4);
(d8) - (dr3 (- r3 xa4 + r3 xa1 + r3 x4 - r3 x1)
+ dr1 (r1 xa4 - r1 xa3 - r1 x4 + r1 x3)
+ dr4 (r4 xa3 - r4 xa1 - r4 x3 + r4 x1))
/((xa3 - xa1 - x3 + x1) ya4 + (- xa4 + xa1 + x4 - x1) ya3
+ (xa4 - xa3 - x4 + x3) ya1 + (- xa3 + xa1 + x3 - x1) y4
+ (xa4 - xa1 - x4 + x1) y3 + (- xa4 + xa3 + x4 - x3) y1)
```

INNACURACY ON Z0

```
(c9) dz01:ratsimp(dz0,x0,y0,z0);
(d9) - (x0 ((dr1 r1 - dr3 r3) ya4 + (dr4 r4 - dr1 r1) ya3
+ (dr3 r3 - dr4 r4) ya1 + (dr3 r3 - dr1 r1) y4 + (dr1 r1 - dr4 r4) y3
+ (dr4 r4 - dr3 r3) y1) + (- dr1 r1 xa3 + dr3 r3 xa1 + dr1 r1 x3 - dr3 r3 x1)
ya4 + (dr1 r1 xa4 - dr4 r4 xa1 - dr1 r1 x4 + dr4 r4 x1) ya3
+ (- dr3 r3 xa4 + dr4 r4 xa3 + dr3 r3 x4 - dr4 r4 x3) ya1
+ (dr1 r1 xa3 - dr3 r3 xa1 - dr1 r1 x3 + dr3 r3 x1) y4
+ (- dr1 r1 xa4 + dr4 r4 xa1 + dr1 r1 x4 - dr4 r4 x1) y3
+ (dr3 r3 xa4 - dr4 r4 xa3 - dr3 r3 x4 + dr4 r4 x3) y1
+ ((dr3 r3 - dr1 r1) xa4 + (dr1 r1 - dr4 r4) xa3 + (dr4 r4 - dr3 r3) xa1
+ (dr1 r1 - dr3 r3) x4 + (dr4 r4 - dr1 r1) x3 + (dr3 r3 - dr4 r4) x1) y0)
/((xa3 - xa1 - x3 + x1) ya4 + (- xa4 + xa1 + x4 - x1) ya3
+ (xa4 - xa3 - x4 + x3) ya1 + (- xa3 + xa1 + x3 - x1) y4
+ (xa4 - xa1 - x4 + x1) y3 + (- xa4 + xa3 + x4 - x3) y1) z0)
```

VALUE FOR THE INRIA PROTOTYPE

DX0

```
(d24) - 8.703684e-03 (3.1499999 dr4 r4 + 8.5519999 dr3 r3 - 11.702 dr1 r1)
```



DY0

(d25) 8.703684e-03 (11.638 dr4 r4 - 4.878 dr3 r3 - 6.76 dr1 r1)

DZ0

(d30) - 8.703684e-03 (dr4 (11.638 r4 y0 - 3.15 r4 x0 - 42.053401 r4)  
+ dr3 (- 4.878 r3 y0 - 8.5520001 r3 x0 - 48.518002 r3)  
+ dr1 (- 6.76 r1 y0 + 11.702 r1 x0 - 24.322477 r1))/z0

```
(c3) /* calculate the stiffness matrix for the medium
position of the manipulator : we have
```

```
    T-1      -1
```

```
    C= J      K J
```

```
where k is the stiffness matrix in the articular configuration i.e.
k*the identity matrix > Here for the MSSM,psi=teta=phi=x0=y0=0
*/
```

```
/* rotation matrix */
```

```
rot:matrix([1,0,0],[0,1,0],[0,0,1])$
```

```
/* position of the mobile */
```

```
(c4) cen:matrix([0],[0],[z0])$
```

```
/* articulation point for the mobile */
```

```
(c5) blr:matrix([-x2],[y1],[0])$
```

```
(c6) b2r:matrix([x2],[y1],[0])$
```

```
(c7) b3r:matrix([x3],[y3],[0])$
```

```
(c8) b4r:matrix([x4],[y4],[0])$
```

```
(c9) b5r:matrix([-x4],[y4],[0])$
```

```
(c10) b6r:matrix([-x3],[y3],[0])$
```

```
/* articulation point for the mobile for the base*/
```

```
(c11) a1:matrix([-xa2],[ya1],[0])$
```

```
(c12) a2:matrix([xa2],[ya1],[0])$
```

```
(c13) a3:matrix([xa3],[ya3],[0])$
```

```
(c14) a4:matrix([xa4],[ya4],[0])$
```

```
(c15) a5:matrix([-xa4],[ya4],[0])$
```

```
(c16) a6:matrix([-xa3],[ya3],[0])$
```

```
(c17) bc1:rot.blr$
```

```
(c18) bc2:rot.b2r$
```

```
(c19) bc3:rot.b3r$
```

```
(c20) bc4:rot.b4r$
```

```
(c21) bc5:rot.b5r$
```

```
(c22) bc6:rot.b6r$
```

```
(c23) u1:cen+bc1-a1$
```

```
(c24) u2:cen+bc2-a2$
```

```
(c25) u3:cen+bc3-a3$
```

```
(c26) u4:cen+bc4-a4$
```

```
(c27) u5:cen+bc5-a5$
```

```
(c28) u6:cen+bc6-a6$
```

```
/* inverse of the jacobian matrix */
```

```
(c29) j01:matrix([u1[1],u1[2],u1[3],bc1[3]*u1[2]-bc1[2]*u1[3],
-bc1[3]*u1[1]+bc1[1]*u1[3],bc1[2]*u1[1]-bc1[1]*u1[2]],
[u2[1],u2[2],u2[3],bc2[3]*u2[2]-bc2[2]*u2[3],
-bc2[3]*u2[1]+bc2[1]*u2[3],bc2[2]*u2[1]-bc2[1]*u2[2]],
[u3[1],u3[2],u3[3],bc3[3]*u3[2]-bc3[2]*u3[3],
-bc3[3]*u3[1]+bc3[1]*u3[3],bc3[2]*u3[1]-bc3[1]*u3[2]],
[u4[1],u4[2],u4[3],bc4[3]*u4[2]-bc4[2]*u4[3],
-bc4[3]*u4[1]+bc4[1]*u4[3],bc4[2]*u4[1]-bc4[1]*u4[2]],
[u5[1],u5[2],u5[3],bc5[3]*u5[2]-bc5[2]*u5[3],
```

```

-bc5[3]*u5[1]+bc5[1]*u5[3],bc5[2]*u5[1]-bc5[1]*u5[2]],
[u6[1],u6[2],u6[3],bc6[3]*u6[2]-bc6[2]*u6[3],
-bc6[3]*u6[1]+bc6[1]*u6[3],bc6[2]*u6[1]-bc6[1]*u6[2]])$
(c30) j01t:transpose(j01)$
(c31) j01t.j01$
(c32) factor(%);

```

/\* STIFFNESS MATRIX: THE RESULT MUST BE  
MULTIPLIED BY THE STIFFNESS OF THE  
DAMPER: rows are included between [[ and ]] \*/

```

(d32) matrix([[2 (xa42 - 2 x4 xa4 + xa32 - 2 x3 xa3 + xa22 - 2 x2 xa2 + x42
+ x32 + x22 )], [0], [0], [0], [- 2 (x4 xa4 + x3 xa3 + x2 xa2 - x42 - x32
- x22) z0], [- 2 (x4 xa4 ya4 - x42 ya4 + x3 xa3 ya3 - x32 ya3 + x2 xa2 ya1
- x22 ya1 - xa42 y4 + x4 xa4 y4 - xa32 y3 + x3 xa3 y3 - xa22 y1
+ x2 xa2 y1)], [[0], [2 (ya42 - 2 y4 ya4 + ya32 - 2 y3 ya3 + ya12 - 2 y1 ya1
+ y42 + y32 + y12)], [- 2 (ya4 + ya3 + ya1 - y4 - y3 - y1) z0],
[2 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y32 - y12) z0], [0], [0]],
[[0], [- 2 (ya4 + ya3 + ya1 - y4 - y3 - y1) z0], [6 z0 ],
[- 2 (y4 + y3 + y1) z0 ], [0], [0]], [[0],
[2 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y32 - y12) z0], [- 2 (y4 + y3 + y1) z0 ],
[2 (y42 + y32 + y12) z0 ], [0], [0]], [[- 2
(x4 xa4 + x3 xa3 + x2 xa2 - x42 - x32 - x22) z0], [0], [0], [0],
[2 (x42 + x32 + x22) z0 ], [2 (x4 ya4 + x3 ya3 + x2 ya1 - x4 xa4 y4
- x3 xa3 y3 - x2 xa2 y1) z0]], [[- 2 (x4 xa4 ya4 - x42 ya4 + x3 xa3 ya3
- x32 ya3 + x2 xa2 ya1 - x22 ya1 - xa42 y4 + x4 xa4 y4 - xa32 y3 + x3 xa3 y3
- xa22 y1 + x2 xa2 y1)], [0], [0], [0],
[2 (x42 ya4 + x32 ya3 + x22 ya1 - x4 xa4 y4 - x3 xa3 y3 - x2 xa2 y1) z0],
[2 (x42 ya4 - 2 x4 xa4 y4 ya4 + x32 ya3 - 2 x3 xa3 y3 ya3 + x22 ya1
- 2 x2 xa2 y1 ya1 + xa42 y4 + xa32 y3 + xa22 y1 )]]])

```

(c3) /\* calculate the stiffness matrix for the medium  
position of the manipulator : we have

$$C = J^{T-1} K J^{-1}$$

where k is the stiffness matrix in the articular configuration i.e.  
k\*the identity matrix > Here for the MSSM and psi=teta=phi=0

\*/

/\* STIFFNESS MATRIX: THE RESULT MUST BE  
MULTIPLIED BY THE STIFFNESS OF THE  
DAMPER: rows are included between [[ and ]] \*/

```
(d32) matrix([[2 (xa42 - 2 x4 xa4 + xa32 - 2 x3 xa3 + xa22 - 2 x2 xa2 + x42
+ x32 + x22 + 3 x0 )], [- 2 x0 (ya4 + ya3 + ya1 - y4 - y3 - y1 - 3 y0)],
[6 x0 z0], [- 2 x0 (y4 + y3 + y1) z0],
[- 2 (x4 xa4 + x3 xa3 + x2 xa2 - x42 - x32 - x22) z0],
[- 2 (x4 xa4 ya4 - x42 ya4 + x3 xa3 ya3 - x32 ya3 + x2 xa2 ya1 - x22 ya1
- xa42 y4 + x4 xa4 y4 - x02 y4 - xa32 y3 + x3 xa3 y3 - x02 y3 - xa22 y1
+ x2 xa2 y1 - x02 y1 - x4 xa4 y0 - x3 xa3 y0 - x2 xa2 y0 + x42 y0 + x32 y0
+ x22 y0)], [[- 2 x0 (ya4 + ya3 + ya1 - y4 - y3 - y1 - 3 y0)],
[2 (ya42 - 2 y4 ya4 - 2 y0 ya4 + ya32 - 2 y3 ya3 - 2 y0 ya3 + ya12 - 2 y1 ya1
- 2 y0 ya1 + y42 + 2 y0 y4 + y32 + 2 y0 y3 + y12 + 2 y0 y1 + 3 y0 )],
[- 2 (ya4 + ya3 + ya1 - y4 - y3 - y1 - 3 y0) z0],
[2 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y0 y4 - y32 - y0 y3 - y12 - y0 y1) z0],
[0], [- 2 x0 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y0 y4 - y32 - y0 y3 - y12
- y0 y1)], [[6 x0 z0], [- 2 (ya4 + ya3 + ya1 - y4 - y3 - y1 - 3 y0) z0],
[6 z0 ], [- 2 (y4 + y3 + y1) z0 ], [0], [2 x0 (y4 + y3 + y1) z0]],
[[- 2 x0 (y4 + y3 + y1) z0], [2 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y0 y4 - y32
- y0 y3 - y12 - y0 y1) z0], [- 2 (y4 + y3 + y1) z0 ],
[2 (y42 + y32 + y12) z0 ], [0], [- 2 x0 (y42 + y32 + y12) z0]],
[[- 2 (x4 xa4 + x3 xa3 + x2 xa2 - x42 - x32 - x22) z0], [0], [0], [0],
[2 (x42 + x32 + x22) z0 ], [2 (x4 ya4 + x3 ya3 + x2 ya1 - x4 xa4 y4
- x3 xa3 y3 - x2 xa2 y1 - x42 y0 - x32 y0 - x22 y0) z0]],
[[- 2 (x4 xa4 ya4 - x42 ya4 + x3 xa3 ya3 - x32 ya3 + x2 xa2 ya1 - x22 ya1
- xa42 y4 + x4 xa4 y4 - x02 y4 - xa32 y3 + x3 xa3 y3 - x02 y3 - xa22 y1
+ x2 xa2 y1 - x02 y1 - x4 xa4 y0 - x3 xa3 y0 - x2 xa2 y0 + x42 y0 + x32 y0
+ x22 y0)], [- 2 x0 (y4 ya4 + y3 ya3 + y1 ya1 - y42 - y0 y4 - y32 - y0 y3
- y12 - y0 y1)], [2 x0 (y4 + y3 + y1) z0], [- 2 x0 (y42 + y32 + y12) z0],
```

$$\begin{aligned}
& [2 (x_4^2 y_{a4} + x_3^2 y_{a3} + x_2^2 y_{a1} - x_4 x_{a4} y_4 - x_3 x_{a3} y_3 - x_2 x_{a2} y_1 - x_4^2 y_0 \\
& - x_3^2 y_0 - x_2^2 y_0) z_0], [2 (x_4^2 y_{a4} - 2 x_4 x_{a4} y_4 y_{a4} - 2 x_4^2 y_0 y_{a4} \\
& + x_3^2 y_{a3} - 2 x_3 x_{a3} y_3 y_{a3} - 2 x_3^2 y_0 y_{a3} + x_2^2 y_{a1} - 2 x_2 x_{a2} y_1 y_{a1} \\
& - 2 x_2^2 y_0 y_{a1} + x_{a4}^2 y_4^2 + x_0^2 y_4^2 + 2 x_4 x_{a4} y_0 y_4 + x_{a3}^2 y_3^2 + x_0^2 y_3^2 \\
& + 2 x_3 x_{a3} y_0 y_3 + x_{a2}^2 y_1^2 + x_0^2 y_1^2 + 2 x_2 x_{a2} y_0 y_1 + x_4^2 y_0^2 + x_3^2 y_0^2 \\
& + x_2^2 y_0^2) ] ] )
\end{aligned}$$

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

11

12

13

14

15

16

17

18

19

20